|  |
| --- |
| Fralysis:  Freudian Dream Analysis Chatbot |
|  |
| 1 6 4 8 7 0  PROJECT SUPERVISOR:  Chris Thornton  Computing for Digital Media BSc  2018/2019 |



# Statement of Originality

This report is submitted as part of the requirement for the degree of BSc Computing for Digital Media at the University of Sussex. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged

# Acknowledgements

Project Supervisor: Chris Thornton

A note of gratitude for the invaluable support and suggestions required in completing this project, despite many extenuating conditions. Thank you.

# Abstract

This paper investigates the feasibility of developing a mobile application that provides an insight into the meaning of an individual’s dreams based on Freudian dream analysis.

The project will:

* Evaluate current techniques used for Natural Language Processing
* Explore possible virtual environments to best simulate a therapy session
* Document the process of developing a mobile application that provides Freudian dream analysis, covering:
  + Classifying user’s dreams from their input
  + Providing a back-end application to perform dream analysis
  + Maintaining a human aspect throughout the session

By the end of this report, you will have a basic understanding of Freud and his view on dreams as well as an understanding of Natural Language Processing. These ideas will be used to implement a mobile app named Fralysis which can converse with a user and interpret their dream.

Fralysis app and documentation can be found on Github [41].

Contents

[Statement of Originality 1](#_Toc7986939)

[Acknowledgements 2](#_Toc7986940)

[Abstract 2](#_Toc7986941)

[1. Introduction 6](#_Toc7986942)

[1.1 Project Aims and Objectives 6](#_Toc7986943)

[1.2 Problem area and Needs of Intended Users 6](#_Toc7986944)

[1.3 Motivation 7](#_Toc7986945)

[2. Professional Considerations 8](#_Toc7986946)

[2.1 Public Interest 8](#_Toc7986947)

[2.2 Professional Competence and Integrity 9](#_Toc7986948)

[2.3 Duty to Relevant Authority 9](#_Toc7986949)

[2.4 Duty to the Profession 10](#_Toc7986950)

[2.5 Ethical Issues 10](#_Toc7986951)

[3. Related Work 11](#_Toc7986952)

[3.1 Dream Analysis 11](#_Toc7986953)

[3.2 Chatbots 12](#_Toc7986954)

[3.3 Natural Language Processing 13](#_Toc7986955)

[3.4 Sentence Structure 15](#_Toc7986956)

[3.4.1 Sentence features 15](#_Toc7986957)

[3.4.2 Sentence Patterns 17](#_Toc7986958)

[3.5 Interface Design Considerations 18](#_Toc7986959)

[4. Background Research 20](#_Toc7986960)

[4.1 Evaluation of Existing Dream Analysis Applications 20](#_Toc7986961)

[4.1.1 iDream 20](#_Toc7986962)

[4.1.2 Dream-e 21](#_Toc7986963)

[4.1.3 Morpheus Dream App 22](#_Toc7986964)

[4.1.4 Comparison 23](#_Toc7986965)

[4.2 Languages and Packages 24](#_Toc7986966)

[4.2.1 Java 24](#_Toc7986967)

[4.2.2 Python 26](#_Toc7986968)

[4.3 Evaluating IDE environments 32](#_Toc7986969)

[4.3.1 Android Studios 32](#_Toc7986970)

[4.3.2 PyCharm 33](#_Toc7986971)

[5. Requirements 33](#_Toc7986972)

[5.1 Functional Requirements 33](#_Toc7986973)

[5.2 Non-Functional Requirements 34](#_Toc7986974)

[5.3 Condensed Requirements 34](#_Toc7986975)

[5.4 Requirements Analysis 36](#_Toc7986976)

[6. Design 36](#_Toc7986977)

[6.1 Logic 36](#_Toc7986978)

[6.1.1 Flow Diagram 36](#_Toc7986979)

[6.1.2 Use case Diagram 38](#_Toc7986980)

[6.1.3 Class Diagram 39](#_Toc7986981)

[6.1.4 Sequence Diagram 45](#_Toc7986982)

[6.2 User Interface 47](#_Toc7986983)

[7. Implementation 53](#_Toc7986984)

[7.1 Programming Language 53](#_Toc7986985)

[7.2 Fralysis’ MVC Architecture 53](#_Toc7986986)

[7.2.1 Model 53](#_Toc7986987)

[7.3.2 View 54](#_Toc7986988)

[7.2.3 Controller 56](#_Toc7986989)

[7.3 Unique Feature 56](#_Toc7986990)

[7.3.1 Maintaining Therapist characteristics 56](#_Toc7986991)

[7.3.2 Extracting User’s Name from Input 58](#_Toc7986992)

[7.3.3 Extracting User’s Dream from Input 59](#_Toc7986993)

[8. Testing 67](#_Toc7986994)

[8.1 Unit Testing 67](#_Toc7986995)

[8.2 Module testing 67](#_Toc7986996)

[8.3 User testing 68](#_Toc7986997)

[9. Evaluation 69](#_Toc7986998)

[9.1 Successes 69](#_Toc7986999)

[9.2 Limitations 70](#_Toc7987000)

[9.3 Overall 71](#_Toc7987001)

[References 72](#_Toc7987002)

[Appendix 75](#_Toc7987003)

[Appendix 1: Ethical Consent Form 75](#_Toc7987004)

[Appendix 2: Project Plan 78](#_Toc7987005)

[Appendix 3: Freudian Symbols 79](#_Toc7987006)

[Appendix 4: Freudian Interpretations 80](#_Toc7987007)

[Appendix 5: Sample Dreams 83](#_Toc7987008)

[Appendix 6: Log 84](#_Toc7987009)

# 1. Introduction

## 1.1 Project Aims and Objectives

This project aims to simulate Freudian dream analysis using a mobile application. The focus of the analysis is recreational; Fralysis is not intended as a treatment for serious psychological issues.

A user will engage in a conversation with Fralysis. Over the course of the conversation, key elements of a user’s dream will be collected. The user’s dream will be analysed using a store of historic dream data. The results of the analysis are fed back to the user.

Users can retrieve dreams from a past session and compare them.

They can take this further with online research or seek professional guidance should they choose to.

## 1.2 Problem area and Needs of Intended Users

There are many types of dreams [2]. Not all of them require analysis. Four common types are:

* Normal dreams: a dream where the dreamer is not aware they are dreaming and where the experience of the reality of the dream does not provoke fear or anxiety.
* Night terrors: a dream where the dreamer screams, experiences great fear, and flails while they are asleep. This type of dream is more common in children.
* Night mares: a disturbing dream that is comprised of negative emotions, such as fear or anxiety.
* Lucid dreams: a dream where the dreamer knows they are dreaming and can control the experiences within the dream.

A normal dream omitting any fear or anxiety poses no harm to the user. Dream analysis for these types of dreams would be for curiosity as opposed to resolving an unconscious conflict. Going to a therapist would be a major investment for curiosity. This app would certainly appeal to this category of user.

The same applies to people who experience lucid dreams. There may not be any lasting benefits of having lucid dreams analysed professionally, but this app can provide a fun, cheap experience for those who are curious.

For people who have experienced night terrors or nightmares and feel that interpreting their dreams will have significant benefits on their health could start with this app. Any real desire to unravel unconscious burdens should be taken to a qualified therapist.

As everyone has dreams of some kind, the intended user has an interest in the Freudian meaning behind their dreams. It should be usable by anyone.

The user is provided with an easy to understand Freudian analysis of their dream, enabling the user to perform further reading if required.

## 1.3 Motivation

The motivation for this project comes from an interest in psychology. This affects everybody to some extent, especially in relation to dreams.

Developing an app that automates an otherwise expensive service will have the potential to widen the audience involved in this outdated, historic area. Although there are many dream interpretation apps, none target Freudian psychology with a focus on learning about his ideas and applying the latest Natural Language Processing ideas.

# 2. Professional Considerations

Although this project is intended for fun, it tightly relates to simulating therapy. Therefore, it will refer to the professional and ethical standards a clinic would follow.

My project aims to provide an external interpretation of a person’s dreams. Given the nature of Freudian psychology, this can disturb some people, especially if they are particularly suggestable.

Below I will go through the BCS Code of Conduct, and state how I will ensure my project meets the required guidelines.

## 2.1 Public Interest

To maintain due regard for public health, privacy, security and wellbeing of others and the environment, I will have to make the purpose of the app very clear: to simulate the response a therapist may provide when interpreting one’s dreams in a fun, educational and experimental way.

These interpretations are by no means scientifically proven, they are filtered from many other interpretations surrounding a similar theme. This should not be taken as a conclusion but allow a user to take interest in the subject and decide whether they should follow up their concerns with a real therapist.

A cap on the sensitivity can be introduced. This would lead the application to become somewhat of an introduction to dream analysis. For example, a leading female character in the dream may refer to the Oedipus Complex. The interpretation should mention how this can translate into a representation of a mother, without going into excessive details.

A major issue to consider is what information is stored from the user. As this app is responsible for comparing dreams from past sessions, a name is required. As this is only to reference a database, it can be made up.

However, the content of the dream is valuable in creating a database of anonymous dreams linked to Freudian meanings to create better results. This will be stated in the terms and conditions of the application.

There is a degree of social influence that can affect the results provided by the application. There is a large difference in how cultures interpret subjects. For example, Eastern cultures are typically far more family orientated than the West, which is more autonomous. Freud’s ideas are Western. This means these influences can make some interpretations invalid depending on the culture. This is something participant research should cover.

From background research, Google Play Store hosts several apps relating to dream interpretation from ‘Persian dream interpretation’ to ‘Biblical dreams’. These would provide different interpretations for the same stimulus. As my project is based around Freud, this will be the root of all interpretations.

## 2.2 Professional Competence and Integrity

As this project can contain sensitive information, complete discretion is guaranteed. Pseudonyms are encouraged. No information outside of dream analysis will be kept on the participants. This is to comply with all legislation.

By evaluating multiple cultures, this aims to reduce any social biases that may offend users. Constant reference to the Freudian root of interpretation will ensure validity. These can be found in Appendix 4.

## 2.3 Duty to Relevant Authority

The research I conduct will only relate to the subject area. This will be done with due care and diligence. To ensure containment of my project, all research will be conducted on Sussex University campus under the circumstances mentioned above.

## 2.4 Duty to the Profession

As this is a project which aims to promote interest in dream analysis, I aim to uphold its reputation by reflecting the outcomes of traditional therapy.

The reputation of the BCS will be considered and maintained throughout the project.

As this field has no exact answers, results should only stand as possible interpretations. Any interpretations resulting in upset will be modified until there are no further issues.

## 2.5 Ethical Issues

The success of this project can only be acknowledged using human testers. This will involve the participant playing with Fralysis. They will be informed that the primary purpose of this is not to leave an impact on their lives but to extend their understanding of Freud’s ideas.

To comply with the ethical standards, it will be made clear that any interpretation of a dream (positive or conflicting) should only be received as a reference, and by no means a conclusion. This reduces risk of harm or distressing thoughts to users.

To ensure the app remains ethical throughout, participants may be asked about what they find discouraging and what ideas should be left to further research if the user wishes to investigate further.

# 3. Related Work

## 3.1 Dream Analysis

The idea of dream analysis goes back to early civilisations, where dreams were associated with prophetic visions [2].

Freud’s ideas were thrown into fruition with his book: The Interpretation of Dreams [3] in 1899. This introduced his unconventional ideas of Wish Fulfilment, Psychosexual Development, the Oedipus Complex, the Psyche and the Unconscious Mind.

Freud believed dreams were the ‘*royal road to the unconscious’* [3]and provide the best means of understanding the mind’s activities. Dream content has two forms: Manifest – literal object in the dream, and Latent – underlying meaning of symbols [5].

Symbols play a key part in Freud’s ideas [4]. He was against universal relationships between manifest and latent content: ‘*they* *have* *private meanings that can only be discovered through the dreamer’s associations’* [6].

The purpose of dream analysis is to determine the latent meaning and use this to address a client’s concerns.

Freudian analysis is no longer mainstream, and a typical course of psychoanalysis can take many years. Therapy sessions involve one–to–one conversation between a client and a therapist.

If dream analysis is to be implemented as a mobile app, an understanding of chatbots is required, as well as the limitations of implementing a historic idea outside the technological domain.

## 3.2 Chatbots

The first major publishing of the idea of chatbots was by Alan Turing in the 1950’s, where he published an article: ‘Computer Machinery and Intelligence’ [7], which spawned the breakthrough idea of ‘The Turing Test’.

The aim of the Turing Test was to distinguish whether a person was either conversing with a person, or a computer. This idea gave birth to AI when trying to answer: ‘Can machines think?’

Between 1960 – 2000 there were many chatbots with varying specialties from Natural Language Processing, therapist impersonations and simulating clients with mental health symptoms [8].

Joseph Weizenbaum developed one of the first chatbots [9.1]: ELIZA (1962) with the purpose of helping people explore feelings and emotions in a therapeutic (free association) fashion: ‘*it has a crucial psychological utility in that it serves the speaker to maintain his sense of being heard and understood*’ [9.2].

Although having some success as being interpreted as a human by some users, it did not pass the Turing Test. It did leave its legacy: ‘*principles used in ELIZA laid a foundation for the structures of chatbots, such as keywords, specific phrases, and preprogramed responses’* [9.2].

Contrary to ELIZA, PARRY (1972) [10.1] developed by Kenneth Colby was a chatbot which simulates a paranoid schizophrenic. Results showed that only 48% of psychiatrists correctly identified PARRY as a chatbot. PARRY and ELIZA can chat together, and the advancement in PARRY clearly affects results [10.2].

ALICE (1995) [11] succeeded ELIZA with better natural language capabilities and received three Lorem awards. It still did not pass the Turing Test.

Since the 21st century, chatbots have progressed substantially, with big companies providing big investment. Sophisticated bots of our era include Apple’s Siri, Microsoft’s Cortana and Amazon’s Alexa. However, these are not developed for conversational purposes, but as agents of convenience – play music, organise calendars and interface with the web.

Recently, Amazon have shown a large dominance in this field, with a desire to expand Alexa to a ‘socialbot’ [12]. This very much involves a machine learning approach, where the program is exposed to conversational data sets and human neural models of cognition and conversation [13].

The bot responds in a way that is most appropriate to the user’s message. This is done by finding patterns in the data set resembling the context best.

As of now, there are still no chatbots which pass the Turing Test [14]. If this were passed, it would be a significant advancement in AI.

Based on above factors and commercial needs, there is strong relevance for chatbots to be used, especially in therapy. From the preliminary examples, factors of successful chatbots involve:

* Efficient use of Natural Language Processing.
* Applied human characteristics.
* Ability to recognise patterns.

## 3.3 Natural Language Processing

Natural Language Processing (NLP) and chatbot evolution are complimentary [15]. Parameter data is taken and compared against a data set (Figure 1). An output is based on the best match.

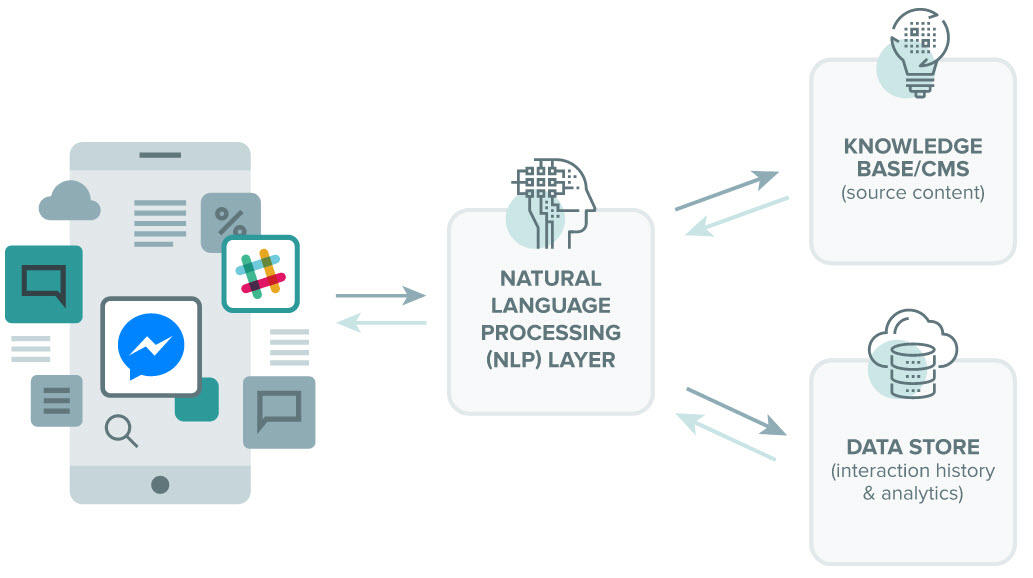


Figure 1 – General NLP Model

Responses are generated by filtering through a data store. The literals are identified by comparing the message to a database containing references to possible contexts.

For example, a user may ask a chatbot about ‘Donald Trump’. The database should have references to ‘Donald Trump’ as a ‘President’ and ’Businessman’. To give this meaning, context should be considered. This is done via a heuristic data store, which looks at the latent meaning regarding the conversation.

For example, ‘I really don’t like Donald Trump’ might have a response: ‘Yes. He seems to be a bad president and a bad businessman’. For the above response to be generated, an emotion – ‘really don’t like’, and direct object – ‘Donald Trump’ must be identified.

In the past, there have been two techniques to do this: Rule-based and Statistical [16]. Rule based traditionally uses nested if-else statements. Even on automating NLP, decision tree algorithms still produced complex conditional rule-based statements.

Statistical models reinvent this by finding the probability of a given context. By analysing a corpus of business documents, references to Trump’s business ventures are likely to appear in a negative context [17].

From above, it is possible to extract a feature from a user’s input and find its emotion by looking at its dependencies. This is vital for dream analysis.

As mentioned in [16]: ‘*Two main issues that affect accuracy are unknown words and ambiguity’.* The advantage of using a chatbot is the ability to lead a user’s input. For this, sentence structure and patterns must be understood.

## 3.4 Sentence Structure

Assuming we can successfully pre - process a user’s input with an NLP toolkit, there needs to be a process of identifying features within the sentence. This requires an understanding of sentence structure and common patterns which are likely to contain features that need to be extracted.

### 3.4.1 Sentence features

In the context of dream analysis, the required features which need to be identified are: subjects, direct objects and compliments.

#### 3.4.1.1 Subject

The subject of a sentence defined as ‘*a noun phrase that functions as the topic of the sentence*’ [18].

For dream analysis, there are three subject classifications that can be made: The user themselves, someone they know or something meaningful like a car or pet. This means every dream will have a subject, regardless of how complex the sentence structure is.

#### 3.4.1.2 Direct Object

Direct objects represent the manifest content of a dream.

For a Freudian analysis, latent content must be attributed to identify the symbolism it represents in the user’s unconscious [5].

For example, Little Hans [19] was a client of Freud’s who had a fear of horses. After some analysis, Freud concluded that Little Hans was suffering from the Oedipus Complex, where the Horse resembled his father. Little Hans was afraid the horse would castrate him.

Using this example, we can identify a subject: “Little Hans”, a subject compliment: “fear” - giving a negative sentiment and the direct object: “horse”. In the given example, horse is symbolic of the Oedipus Complex.

In the context of dream analysis, a direct object in a dream should be any noun which references Freudian dream symbolism. In the example above, the horse is symbolic of the Oedipus Complex.

#### 3.4.1.3 Compliments

Compliments are typically adjectives. They provide sentiment to a sentence. These can either be directed at the subject or the direct object.

For dream analysis, compliments hold information about the sentiment expressed by the subject towards the direct object, which can be applied to its symbolism. This means we can identify whether a symbol in the user’s dream is in a positive or negative context.

For a given sentence with a subject, verb and direct object, it can be said that: ‘the subject applies the verb to the direct object’ [18].

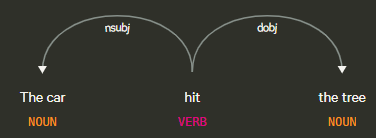


Figure 2 – DisplaCy’s Dependency Visualization Example I [27]

Figure 2 shows an example: ‘*The car hit the tree*’. The subject is the car. The verb is hit. This is often used in a negative context. The direct object is the tree. We can re-phrase the above sentence to say: The subject shows a negative sentiment towards the direct object. This is fine for simple sentences with standalone verbs like hit.

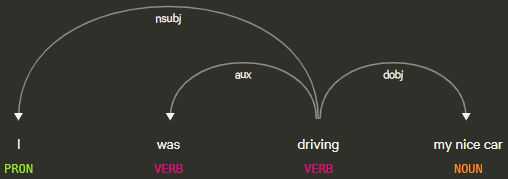


Figure 3 – DisplaCy’s Dependency Visualization Example II [27]

However, take Figure 3: ‘*I was driving my nice car’*. The subject is the user, the verb is driving, and the direct object is car. Driving does not provide a sentiment - it is a neutral verb. The direct object has an object compliment – nice, describing the direct object in a positive context.

For dream analysis, identifying verbs and adjectives gives sentiment to the dream. This gives context to dream symbols.

### 3.4.2 Sentence Patterns

Going back to [18], there are seven common sentence patterns the user can exploit to convey their dream.

The first pattern is known as ’The Be Pattern’, where sentences have the following structure:

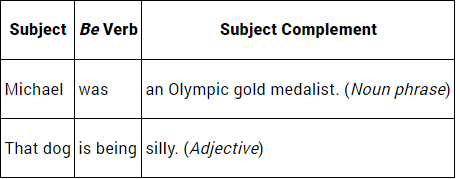


Figure 4 –The Be Pattern

Figure 4 shows sentences starting with subjects. These can either be pronouns, proper nouns or nouns. Pronouns which are most likely to appear in a user’s messages include: ‘I’, ‘I’m’, ‘we’, ‘he’ and ‘she’.

If a message begins with a first-person pronoun, it is logical to assume the subject of the dream is the user.

If a message begins with a third-person pronoun, it is probably not the first message received from a user, but it does suggest the subject of that sentence is referring to someone else. To accurately identify if the subject of a dream is referring to someone else, we can identify if a proper noun is present, as it is unusual to refer to yourself in the third person.

This works in the case of a simple sentence structure like ‘The Be Pattern’ (Figure 4) or ‘The Linking Verb Pattern’. However, it is possible that a ‘Transitive Verb Pattern’ is present (Figure 5 – 6).

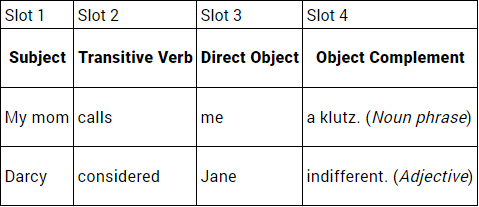
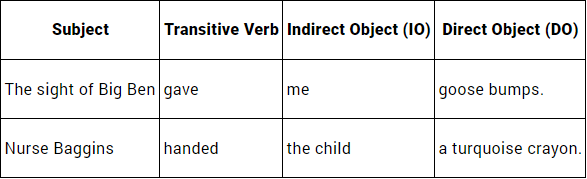
 

Figure 6 - The Transitive Verb + 2 Complements

Verb Pattern

Figure 5 - The Basic Transitive Verb Pattern

For dream analysis, this gives the foundation of which information to extract from a sentence and how to classify it. It suggests that the implementation of the chatbot logic layer must recognise required features despite the message structure of the user. Furthermore, it suggests how to structure the chatbot’s responses to lead the user to provide specific information in their input.

## 3.5 Interface Design Considerations

Mobile applications primarily take two inputs from users: strings and triggered events. Conversational UI [19.1] tries to simplify this further.

With the possibilities of NLP, there must be considerations into how best to interact with the user to simulate a therapy session as accurately as possible. Interfaces affect data validity. Although chatbots traditionally use a message interface [19.2], VR and robotics are the future of UI, and will likely produce different results.

For the given scenario: A user wishes to analyse their dream from the comfort of their home. Typing inputs will certainly take way from the importance of free association [21]. Speech – to – text would be more suitable.

Furthermore, VR can be used to better immerse the user into the dream analysis. This would give better quality of input data.

This is not always the case. Jeff Hawkins, founder of ‘Palm, inc.’ and now involved in applying neuroscience to computing, found: ‘*voice control is unsatisfactory and uncompelling*’ [22]. Having users speak to their phone about sentimental information can be discomforting and may compromise the validity of the users response.

Considering the above, the app development stage will have a strong focus around User Centred Design. The best means of providing information to a chatbot will be explored.

A Model View Controller (MVC) architecture means the same Freudian data model and NLP controller can be applied to various views. The preferred view for development would be through the terminal. The MVP should include a standard mobile messenger interface to enable a user to converse with the chatbot. Given more time and technology, a VR domain can be implemented to focus on user immersion.

This architecture promotes privacy when using the app in various environments. Encapsulating the logic layer in a controller means you can easily update the NLP resources as they evolve. This is the same for the Freudian model, which can be expanded over time using machine learning. Testing confirms the components of the architecture are behaving as they should.

The output of the analysis will be a significant factor. Dreams are often ambiguous. The best method of conveying the post - processed information will be in a readable and descriptive string variable.

# 4. Background Research

## 4.1 Evaluation of Existing Dream Analysis Applications

There are no Freudian dream analysis apps. However, there are a dozen dream apps.

### 4.1.1 iDream

iDream [23] is an app for IOS which acts as an interactive dream dictionary. It allows a user to create dream journals. They can access the dream dictionary directly to look up specific parts of their dream.

For example, going to see a comedian and feeling envy for the quality of their jokes. The app provides an interpretation for a comedian: represents our childish instincts for a lack of seriousness. Envy can represent a jealous desire for something. iDream interface is shown in Figures 7 – 8.

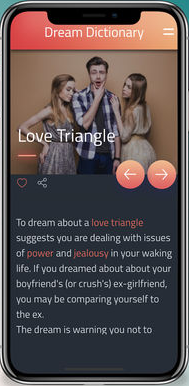
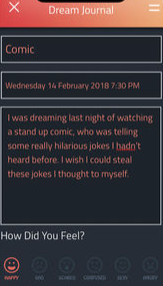
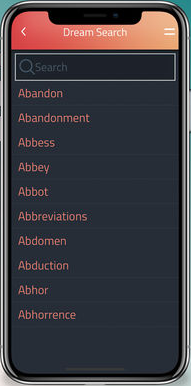
  

Figure 8 – Dream Entity Definition

Figure 7 – Dream Dictionary and Journal

Fralysis differs from this because it provides interactivity between the user and the Model. iDream uses a fixed lookup dictionary for analysis. Fralysis will improve on this by ‘interviewing’ the user and using NLP to filter results. This enables personal responses, which is a requirement from [6].

### 4.1.2 Dream-e

Dream-e [24] takes a different approach from iDream. The app is presented as a virtual therapist and looks like a standard messaging app. The users respond to questions presented by a chatbot.

The app stresses it is not a dream dictionary, but a dream image analysis app. It will find out the therapeutic meaning of the images mentioned in your dream.

Dream-e is a generic dream analysis app. Fralysis aims to focus on Freudian psychology. Although Dream-e is a messaging interface, there is minimal human feel to its messages. Fralysis intends to mimic a Freudian therapist and teach people about its ideas.



Figure 9 – Dream Entity

Figure 10 – Dream Subject

Figure 11 – Chat

After experimenting with Dream-e (Figures 9 - 11), it was clear that the novelty of the chatbot was just to interface with the user to extract symbols of their dream and provide a bespoke dictionary definition. This is much like iDream.

Minimal NLP is involved. The app is not very satisfying, but it does provide a visually appealing GUI.

### 4.1.3 Morpheus Dream App

Morpheus Dreams [25] is an upgrade over iDream with a quality interface. It primarily uses a dream dictionary to dynamically classify the main features of a dream.

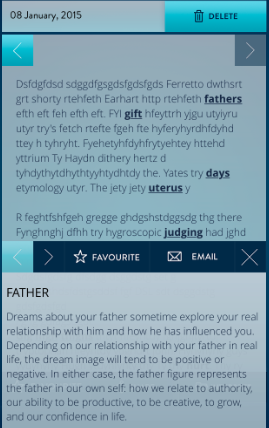
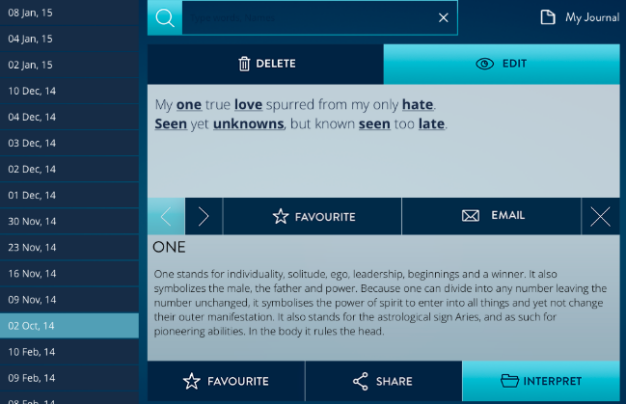
 

Figure 12 – Dream Journal

Figure 13 – Dynamic entity definition

A clever feature is the implementation of the dream journal (Figure 12). As a user records their dream, key features are automatically interpreted (Figure 13). This is a clever use of Natural Language Processing.



Figure 14 – Morpheus’ capabilities

The app has a greater range of functionality. It facilitates alarms and calendars (Figure 14) to motivate usage of the app and build a profile around the user.

If the functionality of Morpheus was implemented as a chatbot, this would be a great interface to Fralysis. This would enable past dreams to be compared and referenced in a therapy session. It paints a picture of what Fralysis should ideally be capable of doing.



Figure 15 – Dream App Comparison

Figure 15 compares current dream analysis apps on the market. It shows some of the features that should be attempted to compete with the current market.

### 4.1.4 Comparison

Overall, the above apps approach dream analysis from different perspective.

All three apps approach dream analysis using the idea of dream dictionaries: iDream and Morpheus using dream journals. Dream-e uses a chatbot to interface with details of the users dream.

From a UI perspective, Dream-e provides the best interface. It gives a viable approach to interacting with a user, although the logic could be improved.

The key difference with these applications and Fralysis is the idea of dream symbols. Fralysis should uphold Freudian values, and Freud had strong ideas about universal dream symbols [6].

To overcome this, an extensive model will have to consider key Freudian ideas. The logical processing will involve simulating a therapy session as closely possible. There will be multiple levels of the interview process. Multiple dreams should be analysed for patterns.

Fralysis analysis should associate the manifest dream content to a latent interpretation based on a Freudian model. This should be paired with an object compliment and targeted towards the subject of the dream.

## 4.2 Languages and Packages

There are many computing languages which have libraries specialising in areas which will be required for producing this app. What follows will consider the certain aspects that must be implemented. These include:

* Communicating in human – like fashion with the user.
* Pre – processing of users’ input via a natural language toolkit.
* Analysing features and sentiment of users’ input using sentence patterns.
* GUI to implement messenger and allow user interaction.

Each section will contain a possible language to provide this. This is dependent on available toolkits. This section is structured so the most valuable language is considered last.

Android applications are typically developed in Java using Android studios. However, this may not be the best solution for this project.

### 4.2.1 Java

Java has a long running history. Initially developed for interactive television but was ‘*too advanced for the digital cable television industry at the time*’.

Popularity came to Java when Java applets became incorporated in web browsers via the Java virtual machine. This meant Java apps could run on any platform by compiling them through an intermediate representation called ‘Java bytecode’.

#### 4.2.1.1 Scanner

Java has a class called Scanner, which belongs in the util.java package. This parses primitive types and strings via regular expressions. Inputs are split into tokens, which is required for NLP.

Scanner will allow a user to enter a dream into a terminal and is useful in extracting inputs at a development stage. Due to its long running history, it is a reliable library.

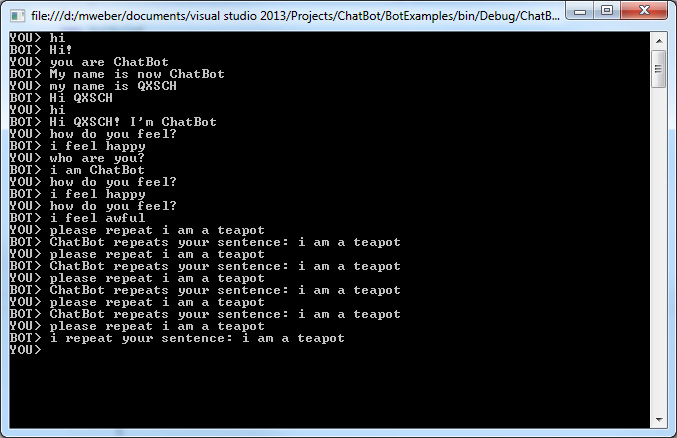


Figure 16 - Chatbot using Scanner

One limitation is that it is not very efficient. It does not provide any text pre - processing. This brings in the need for a more sophisticated tool.

#### 4.2.1.2 JavaFX

Despite being somewhat outdated, JavaFX is a key component of the Java SE environment. JavaFX was created to take over Swing, which was the original means for creating a GUI for Java applications.

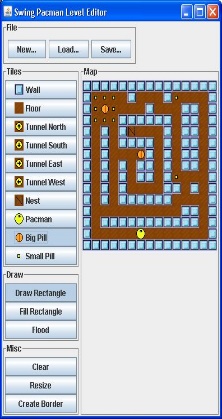


Figure 17 - Swing App

The Swing class extends JFrame – the window that contains UI. JPanels add components to the JFrame. JavaFX uses a theatrical approach of having a stage class representing the screen, and various scene classes handle the components making up the UI.

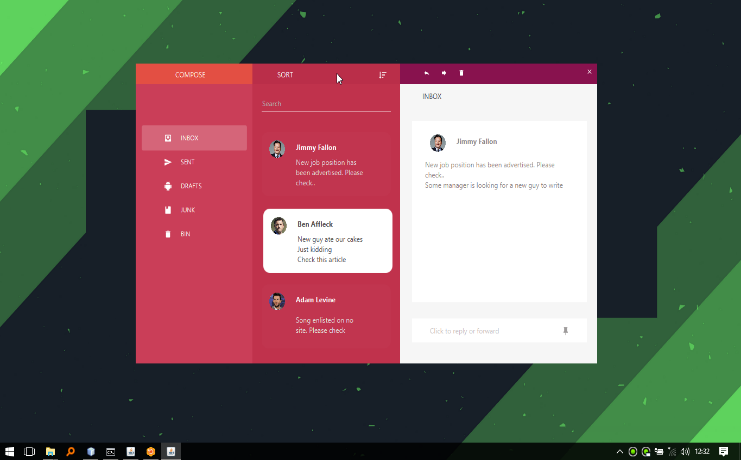


Figure 18 - JavaFX App

Java is multi-platform, and JavaFX combines the JDK and JRE on supported systems. Although there is no official Oracle distribution for JavaFX on Android, you can use the ‘JavaFX-Android SDK’ to create an Android project based on your JavaFX Application.

### 4.2.2 Python

Python is one of the most popular languages to date. It has a broad use, from desktop and web-application development to sophisticated scientific data analysis - both of which are required for the project.

#### 4.2.2.1 Pandas

Pandas is an open source library used for data analysis and generating visual representations of results. Dataframe objects (Figure 19) allow for manipulating data.

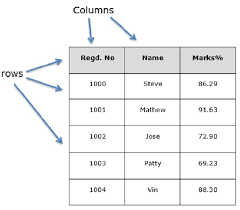


Figure 19 - Pandas Dataframe

Implementing this correctly helps support logic validity. Patterns are visually recognisable as shown in Figure 20. This can be used to visualise dream features and their associated symbolism; instead of providing a dream interpretation as text.



Figure 20 - Pandas Graph

If Fralysis were to expand from a chatbot, it could provide visualisation of a user’s dreams and which symbols and sentiments are recurring.

#### Natural language toolkits:

For a general description of NLP, go to 3.4.

Python provides several well supported NLP libraries. These can be used for identifying features of a sentence.

Given Java’s age over Python, it has been in the NLP space for longer [26]. However, Python dominates this field. Therefore, this section will focus on Python libraries.

Using an NLP toolkit will save the time required to manually classify all word types that may appear in dreams. The two toolkits I will look at are NLTK and SpaCy.

##### 4.2.2.2 NLTK

NLTK is a package for symbolic and statistical NLP. It was developed to aid research and teaching in this field by providing empirical linguistics, cognitive science, information retrieval and machine learning – all relevant to the project.

The main features it supports are part-of-speech tagging and chunking.

Part-of-speech tagging involves marking up words in a sentence based on their definition and the context. For example, clowns are typically seen as funny to some people but scary to others.

NLTK provides the means to analyse words based on their context as opposed to stereotypes – which would be the biggest cause of misinterpreting dreams.

A limitation of NLTK is that it only supports the English language. It is also predominantly a research tool for study purposes, providing only a basic platform for prototyping and creating research systems.

For this reason, SpaCy seems to be a better candidate.

##### 4.2.2.3 SpaCy

SpaCy [27] is another open source library that provides advanced Natural Language Processing features. It is designed for industrial strength word processing and is handled via a pipeline.

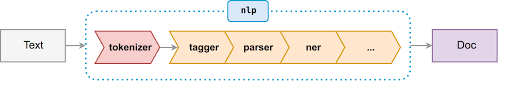


Figure 21 - SpaCy Pipeline

The pipeline works by taking a text document and tokenising it as illustrated in Figure 21. As documented by SpaCy, tokenisation: *‘Segments text into words and punctuation and creates a Doc object with the discovered segment boundaries’* [28].

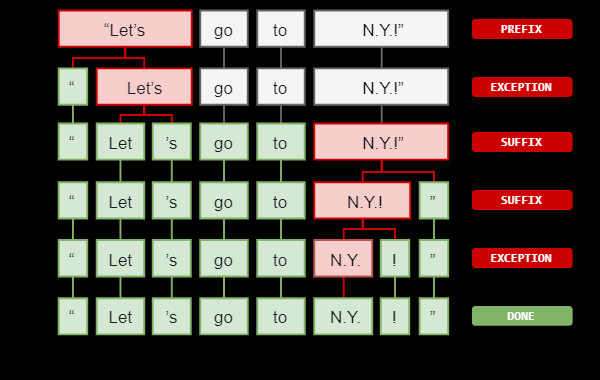


Figure 22 - sentence words to tokens

The tagger [29] gives words attributes. This describes properties like a part of speech, lemma (Figure 22) and whether it is representative of an entity – person, organisation, location and so on.

The tagger is initialised using a neural model (‘thinc.neural.Model’) which is essentially a data-training file: *‘A machine-learning algorithm is a mathematical model that learns to find patterns in the input that is fed to it. This input is referred to as training data’*. [13]

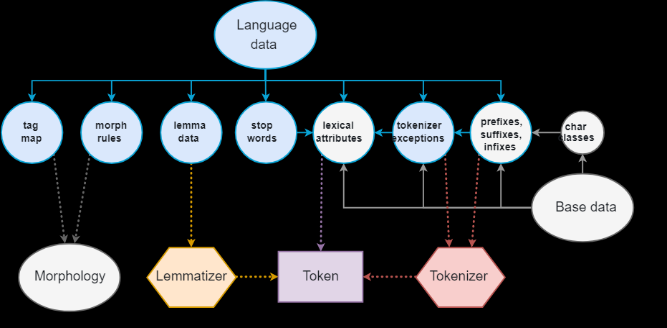


Figure 23 - Using a data file to categorise a token

Because the tagger is a machine learning algorithm, it is possible to replace the model and train your own tagger on specific data. If the application was further developed to support features like the history of Freud, it may be required to train the tagger on specific Freudian papers and books for more accurate associations. This can be done by changing the language data in Figure 23.

It is fair to assume training an NLP pipeline on corpuses from the 19th and 20th century would certainly provide different results. The default model used by SpaCy means: ‘*The parser and NER use an imitation learning objective to deliver accuracy in-line with the latest research systems, even when evaluated from raw text’* [30].

Fralysis requires name identification, as well as sentence features. SpaCy provides this through its part of speech attribute, where a name is typically associated with a proper noun. Sentence features mentioned in 3.4.1 can be extracted in a similar way.

Other domain advantages of SpaCy is its support for multiple languages including German, Spanish, French, Italian and Dutch. Additionally, it supports multi-language Named Entity Recognition. This is important for modern chatbots and improves the user experience.

#### GUI Libraries:

##### 4.2.2.4 KIVY

KIVY [31] is a GUI library for Python. KIVY allows apps to be packaged for multiple platforms easily with Buildozer [39].

Where an app is a chatbot, the requirements from the GUI are straightforward: it must provide a chat screen where the chatbot can send messages to the user, and the user can respond.

Using an MVC design means the model can be as complex as required, but ultimately a string is returned. The role of KIVY in the app is to provide a messenger view. The bot’s message is written to the display. The user responds to this. The screen is updated, and the message is passed to the model to generate the next chatbot response.

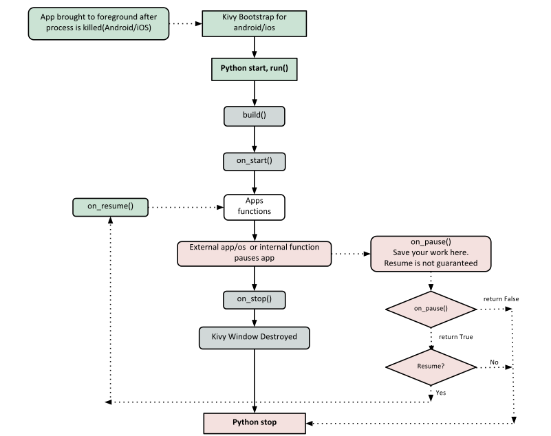
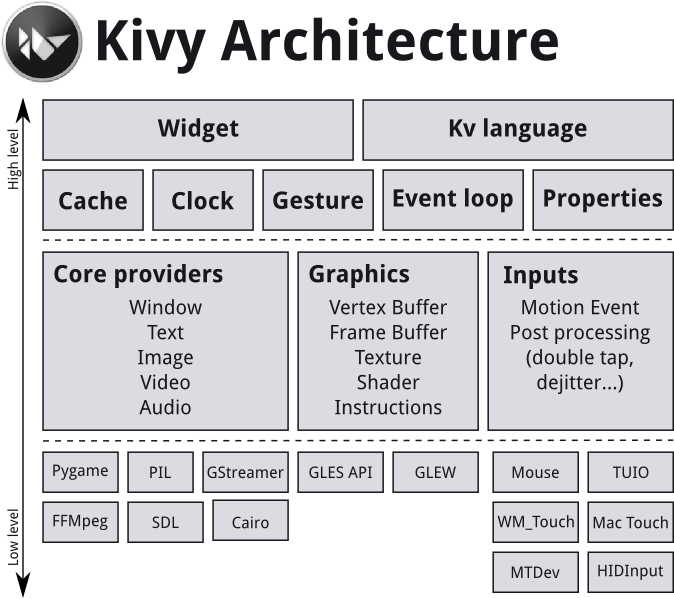
 

Figure 24 – KIVY Flow Diagram

Figure 25 – KIVY architecture

KIVY allows for a minimal viable implementation for a working app. Ideally, this project could expand to explore different types of interfaces. For example, moving away from typing to speech-to-text input (Also supported by KIVY).

KIVY is ideal for offline mobile use. The program can be packaged up for IOS, Android and Desktop.

However, given the nature of machine learning, file sizes can be very large. A solution would be to set the program up on a server and pass inputs via a web-based interface.

##### 4.2.2.5 HTML

A web app could be used as a universal GUI. Html and php can be utilised to communicate with the Python server. This still implements the MVC model.

Strings are read from the web app view like that in Figure 26. These are passed to the backend logic controller. A string which represents the chatbots message is returned to the view.

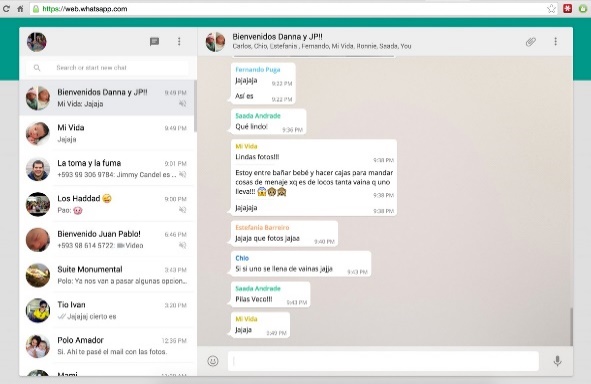


Figure 26 – Whatsapp Web App

This approach should be considered if expanding Fralysis. As offline usability is required for the MVP, KIVY is currently the library of choice for the MVP.

## 4.3 Evaluating IDE environments

IDE environments help organise projects and visualise interfaces.

### 4.3.1 Android Studios

Android studios is a standard development platform for creating android applications.

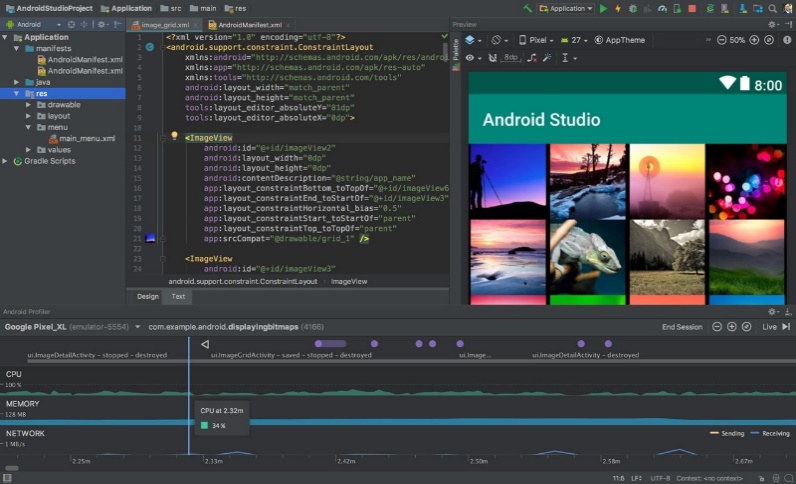


Figure 27 – Android Studios

It has a plethora of libraries to structure the application layout. The frontend is typically written in XML. A great benefit is visualising the interface dynamically.

### 4.3.2 PyCharm

For Python development, PyCharm is the main IDE.

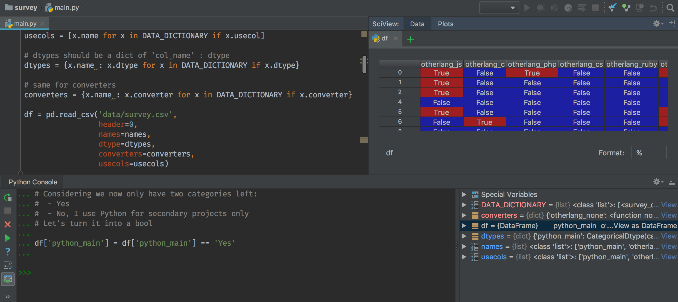


Figure 28 - PyCharm

It resembles android studios and provides a good workflow between various classes and libraries.

PyCharm will most likely be used to develop Fralysis.

# 5. Requirements

## 5.1 Functional Requirements

Fralysis should have a messenger interface. It should prompt a user to converse with a therapist as if they were a client. The individual is encouraged to provide a pseudonym and describe their dream. If a name is recognised in the database, previous dreams of the user will be retrieved.

The controller should process a user’s message. If a name is provided or a dream is described, this should be classified correctly. The controller generates messages based on the classification. Patterns in accordance to the Freudian model will be identified. The model should contain at least 100 references to common manifest dream contents [5]. This should reference a possible latent content description for its Freudian meaning. It is also responsible for categorising sentence features such as compliments [18].

The logic layer will check to see if all required features have been collected and produce required requests for missing information.

Once an identified user has provided a complete dream, a choice will be given to provide more dreams or to get an analysis.

Analysis should start by referencing the subject of the dream. It will then provide an overall sentiment of the dream based on the subject compliment. It will then address which Freudian symbols have been classified. Identified manifest content is described using the latent content description found in the model.

A final reference to the Freudian argument should be provided to facilitate further research. Responses from the chatbot therapist should consistently resemble a human.

## 5.2 Non-Functional Requirements

Responses should be generated within 2 seconds. This should not be instantaneous. 2 second responses provide realistic thinking time.

The app should be written in Python. The controller should be implemented using SpaCy. The interface should be implemented with KIVY.

The application should run on multiple mobile devices. This means packaging the app for IOS, Android and Windows. Scaling should be automatic based on the device display.

## 5.3 Condensed Requirements

|  |  |  |
| --- | --- | --- |
| Functional Requirements | | |
| Ref | Description | Mandatory/Desirable |
| F1 | Provide a textual - messaging application interface. | Mandatory |
| F2 | Provide text-to-speech functionality. | Desirable |
| F3 | Prompt user to provide name. | Mandatory |
| F4 | Identify if user name is already in the database and retrieve previous user dreams. | Desirable |
| F5 | Prompt users to describe their dream. | Mandatory |
| F6 | Classify key features of a dream. | Mandatory |
| F7 | Identify missing features from a dream. | Mandatory |
| F8 | Identify a complete dream | Mandatory |
| F9 | Provide analysis for a complete dream | Mandatory |
| F10 | Suggest an application to real world situations. | Desirable |
| F11 | Refer to original Freudian argument | Mandatory |
| F12 | Output response in textual form | Mandatory |
| F13 | Output response in auditory form | Desirable |
| F14 | Provide a dream dictionary in Freudian style. | Desirable |
| F15 | Visual construction of the dream. | Desirable |
| F16 | Compare Dreams | Mandatory |
| F17 | Android compatible | Mandatory |
| F18 | IOS and Windows compatible | Desirable |
| F19 | Handle ambiguous inputs | Mandatory |
| F20 | Session should always terminate; does not have to be successful | Mandatory |
| Non-Functional Requirements | | |
| NF1 | Written in Python. | Mandatory |
| NF2 | Product of TDD | Mandatory |
| NF3 | Enforce MVC architecture | Mandatory |
| NF4 | Support Android version up to 7.1.1 | Mandatory |
| NF5 | Have a repository on Github | Mandatory |
| NF6 | Provide at least 100 dream interpretations. | Desirable |
| NF7 | Use SpaCy for NLP | Desirable |
| NF8 | Use KIVY for GUI | Mandatory |
| NF9 | Average response times within a second. | Desirable |
| NF10 | A human therapist should be simulated. | Mandatory |
| NF11 | Randomise messages | Mandatory |

**Figure 29 – Requirements Table**

## 5.4 Requirements Analysis

The target audience for this project will be interested in Freudian psychology and desire an insight in to what their dreams represent in this context. This is implied through F5, F6, F10 and F11. An ideal system will ensure the result of F6 is factually accurate. It will also include an accurate representation of F14.

I will deem this project successful if the user has an enjoyable and informative experience interacting with the simulated Freudian therapist – NF10.

The app will be designed for smartphones. NF1 via NF8 and NF4 need to be fulfilled.

NF6 will determine the scope of the app and how detailed it is. The ideal product would contain an interpretation for every possible dream. This isn’t an achievable requirement as it would involve an excessive amount of storage and time. A hundred interpretations provide an acceptable scope.

In summary, implementing the requirements will result in an application capable of taking user inputs for a GUI view, determining a meaning based on the model and outputting a valid Freudian analysis.

# 6. Design

## 6.1 Logic

The look and function of the app has been designed, based on background research.

### 6.1.1 Flow Diagram

Figure 30 illustrates the logic required to extract user dream information. It shows checkpoints needed for extracting information from a user.

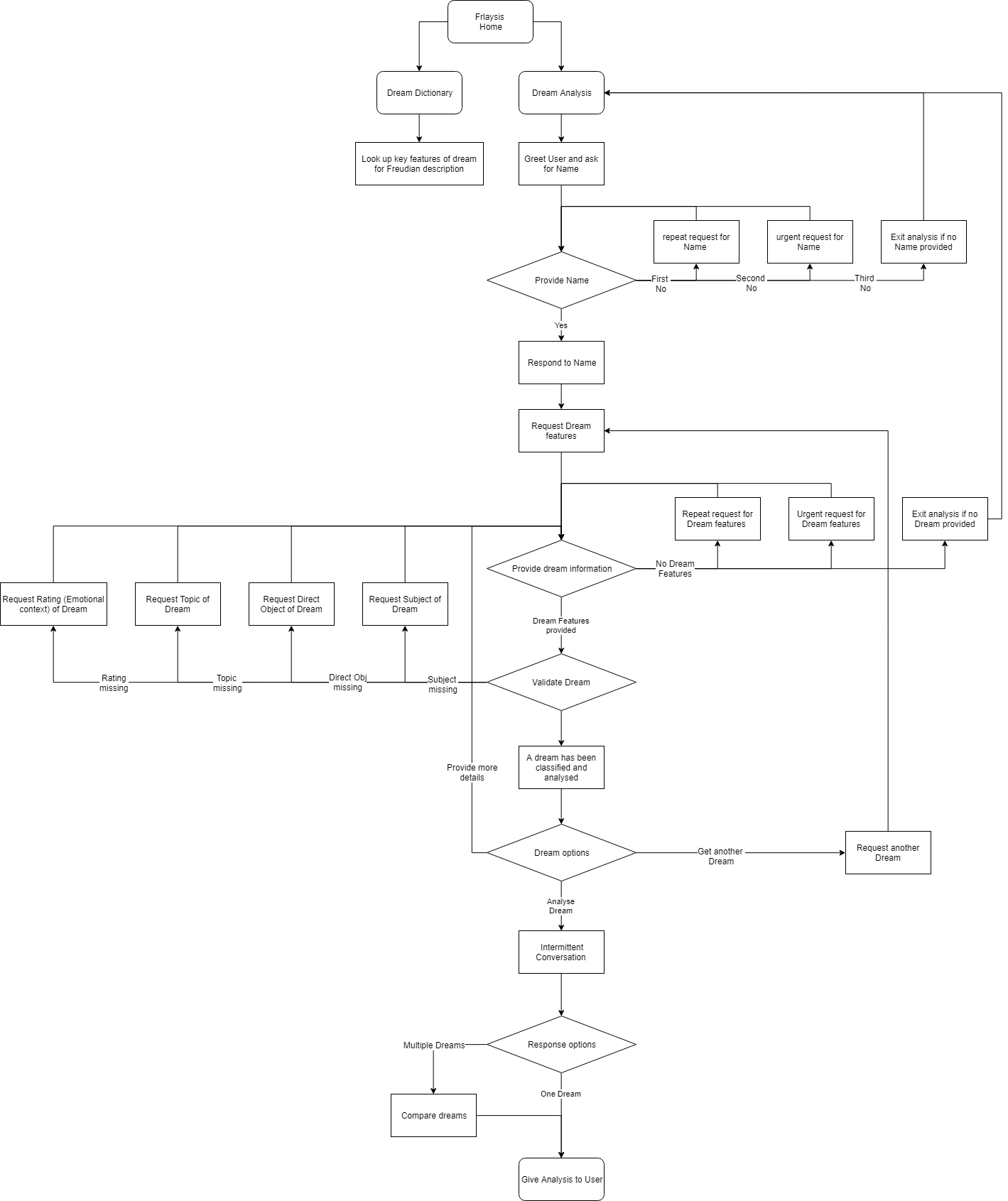


Figure 30 – Flow Diagram

Fralysis begins life in the home screen. This will provide a button to navigate to the Dream Dictionary or the Dream Analysis Chatbot.

The Dream Dictionary provides an interface to the data model, where the manifest object in a dream is coincided with a Freudian latent definition [5].

The Dream Analysis page holds the chatbot logic. The user is greeted. This simulates a therapy session introduction. If a name has not been provided, this will be requested. This simulates a real conversation.

The chatbot requests details about the user’s dream. This searches for features in the input referred to in 3.4.1. The app should only proceed once the user has provided information on all the necessary features.

If the user does not provide the requested information, levels of repeat requests range from mild – ‘sorry, could you repeat that’, to urgent – ‘If you don’t tell me your name, I am going to have to kick you out!’. This gives human appeal.

Once a dream is collected, the user should be asked if they wish to provide another dream or receive an analysis. Providing a new dream involves going back and requesting another dream. Getting an analysis gives feedback to the user and ends the therapy session.

Encapsulating the logic like this enables code reuse. This is useful when transitioning from a standard chatbot to a more immersive VR environment.

### 6.1.2 Use case Diagram

Figure 31 shows which features are imperative from our session entities.

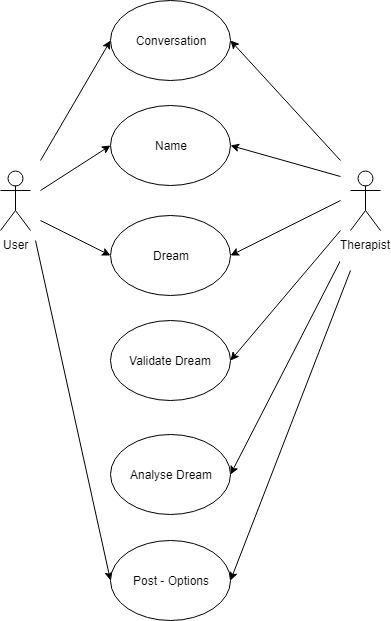


Figure 31 – Use Case Diagram

The required interactions are very minimal.

The user replies to the therapist’s messages.

The therapist has the responsibility of validating the user’s inputs. This is done by extracting features of their input and building up their dream until an analysis can be made.

Given these responsibilities, a class diagram can be designed.

### 6.1.3 Class Diagram

Figure 32 gives a lower level view of the classes, methods and variables required in our program to achieve the required functionality.

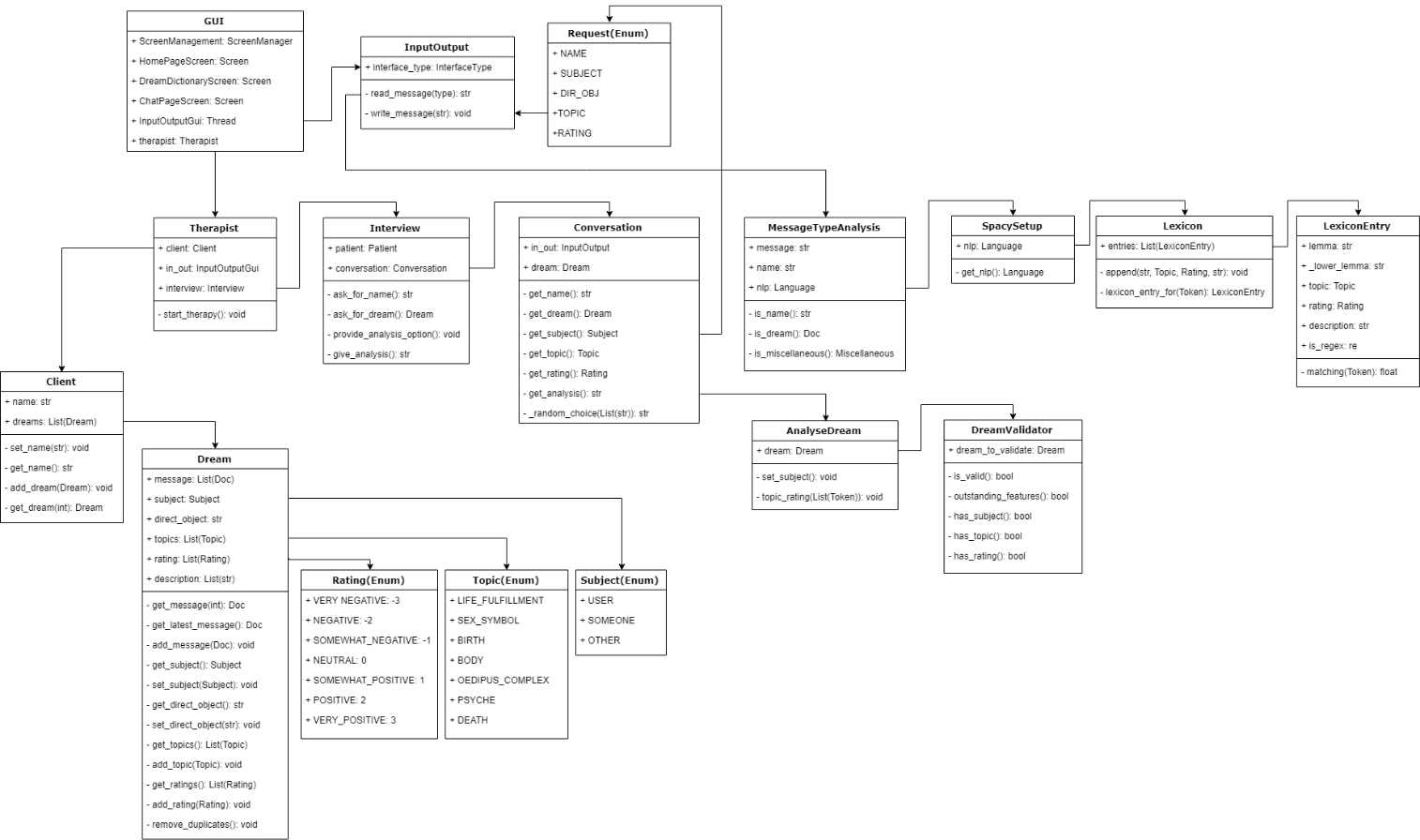


Figure 32 – Class Diagram

#### 6.1.3.1 GUI

The KIVY library (reference in 5.2) implements a GUI. This contains a Homepage Screen, Dream Dictionary Screen and a Dream Analysis Screen. These are controlled by a ScreenManager.

The user can talk to the chatbot by navigating to the Dream Analysis Screen from the Homepage Screen.

The GUI lets the user talk to the therapist via a messenger interface. This could be considered a container for the therapy session, which is hosted by the Therapist.

#### 6.1.3.2 Client

A client is the user. They should have a unique pseudonym - allowing a user’s dream history to be retrieved. It also provides a reference point in the chat.

Clients also have a list of dreams. These should be complete dreams and are used by the Therapist to provide a dream analysis.

#### 6.1.3.3 Dream

A dream is defined much like a fictional story (Figure 33).

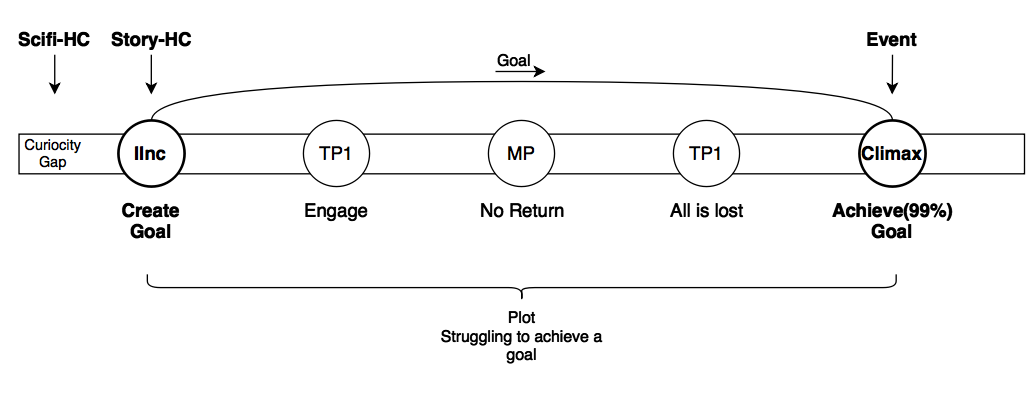


Figure 33 – Story Structure

Stories have a main character – this gives perspective. A main character referred to in the subject of the sentence. This allows an interpretation to be directed to the context of the user.

Using the propositional logic from 3.4, we can assume the subject compliment is the same as the object compliment. This links the sentiment of the dream’s latent meaning to the subject.

With the features above, a dream is given to a therapist, who can take its features and feeds them back to the user.

Take Little Hans [19] who had a phobia of horses. He thought horse would try to castrate him. The subject is Little Hans. The direct object is the horse. Horse has a negative compliment – castrate.

Through the therapy session, Freud associates the horse to Hans’ father, symbolising the Oedipus Complex. Bringing this to Hans’ consciousness eradicated his phobia of horses. A similar case was found with Freud’s former client – Anna O: ‘s*ymptoms disappearing with awareness of their unconscious preconditions has been confirmed by all subsequent research’* [32].

#### 6.1.3.4 Therapist

The therapist controls the session and has clients.

The therapist begins the therapy by starting an interview with the client. The therapist is responsible for terminating the session.

#### 6.1.3.5 Interview

The interview provides an interface for the flow diagram described in Figure 27. It controls when to move onto the next checkpoint.

Interview works by initiating one or more conversations and determines when the conversation cycle is complete – once the requested information is identified.

A complete interview is when there is no more information required for analysis.

The process of talking with the client is handled in the conversation.

#### 6.1.3.6 Conversation

The conversation writes messages to the GUI prompting responses from the user. The purpose of this class is to imitate a human therapist.

This is done by wrapping information requests in styled responses. This is straight forward for greeting a client – ‘Hello there! How are you?’. Requesting a name and dream follow the same format.

However, requesting specific information in an indirect way requires therapist specific responses – ‘Lets develop your dream a bit more. How did the spider make you feel?’. The corresponding user’s reply will hopefully contain a subject compliment (3.4.1).

Another therapist styled request: ‘How did the spider appear in that context?’ is likely to lead to a response containing an object compliment.

As there are likely to be multiple conversations in a session. Multiple message structures should be provided for requesting features. This reduces the predictability of the therapist’s response.

As well as message structures, there are request levels. These range from an initial request, to a repeat request and finally an urgent request. A message may not contain recognisable features. This gives the impression that the therapist is becoming frustrated with the user’s responses. It is also a tool to promote simpler user responses.

Assuming the user is giving valid responses, something must classify the message to inform the conversation that a name or dream features have been identified.

#### 6.1.3.7 MessageTypeAnalysis

If the user provides their name, a proper noun should be identified in their input. If it is, the name is returned to the conversation, passed back to the interview, and the client’s name is set.

If the user’s input contains features of a dream, the entire message is returned to the conversation for further analysis. The input is stored in the dreams list of messages and given to AnalyseDream.

Before the message is returned, it is pre-processed by SpaCy language object and converted to a Document object.

#### 6.1.3.8 SetupSpacy

Details on NLP implementation are provided in section 8.

SetupSpacy is responsible for initialising the pipeline that prepares a user’s message for analysis.

It starts by taking a SpaCy language object and initialising it with a training data set as described in 3.3.

#### 6.1.3.9 Lexicon

A lexicon is ‘*the vocabulary of a person, language, or branch of knowledge’.* Freudian specific terms such as ‘mother’ and ‘house’ can be added with a Freudian symbol – ‘Oedipus Complex’ and ‘body’. These will have a Freudian description.

Multiple levels of analysis can be made like this.

The highest level references the dream context. References to symbolism and their sentiment follows. Finally, manifest content can be illustrated alongside its latent meaning based on the lexicon.

The lexicon is used to extend the SpaCy pipeline. If a user’s input matches the lexicon, it is analysed by AnalyseDream.

#### 6.1.3.10 AnalyseDream

This class takes a dream object. It looks at its latest message. Features are extracted and added to the dream.

If the dream does not have a subject, it will attempt to identify one (using rules from 3.4.1). Each dream should have one subject.

If a direct object is found that represents a manifest content, the latent symbol should be added to the dream’s topics.

If a compliment is found, this should be added to the dream’s ratings. Details on how this is done is provided in the section 8 of this report.

Once the dream object has been updated with the features found in the message, it needs to be validated.

#### 6.1.3.11 DreamValidator

This assesses a dream. It takes the dream object and checks its contained features.

A list of missing features is generated and passed back to the conversation to be requested.

If there are no missing features, the dream is valid.

By integrating the loosely coupled, highly cohesive classes above, a sequence diagram can confirm logical validity.

### 6.1.4 Sequence Diagram

Figure 34 shows the low-level interaction between objects in Figure 32 to implement the flow logic in Figure 30.



Figure 34 – Sequence Diagram

Figure 34 gives a more detailed view on how the classes interact with each other, simulating the interaction between a therapist and a client.

It illustrates the process of requesting a name as well as requesting and validating a dream. An iterative cycle is shown for repeating requests in case user information is not relevant.

## 6.2 User Interface

The interface can be modelled to get an idea of how the app will look, and which colours compliment the idea of dream analysis - they should be subtle and calming.

Figures 35 – 51 are prototypes showing how a user will interact with Fralysis. They illustrate how the final product should look and feel. Designs were created in Android Studios in XML on an Android VM.

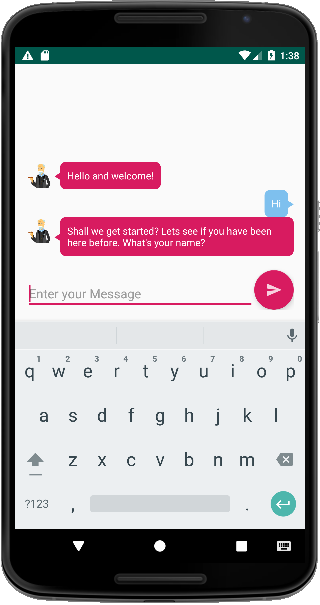
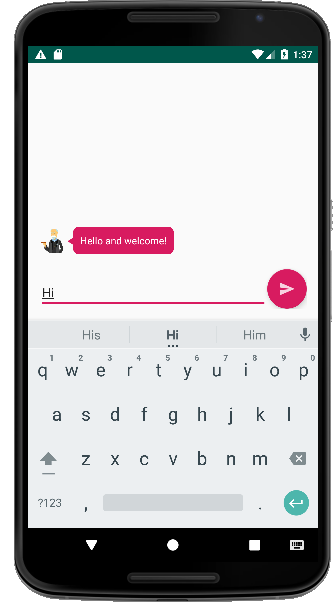
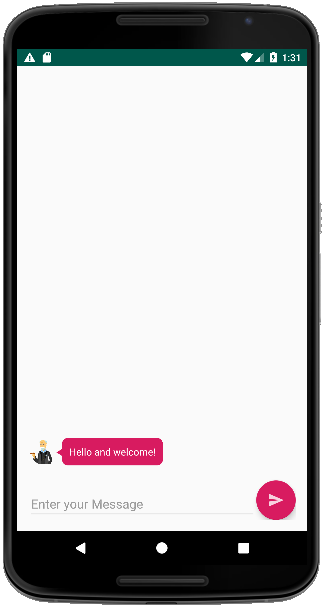


Figure 37 – Greeting Response

Figure 36 – User Input

Figure 35 – Introductory Message

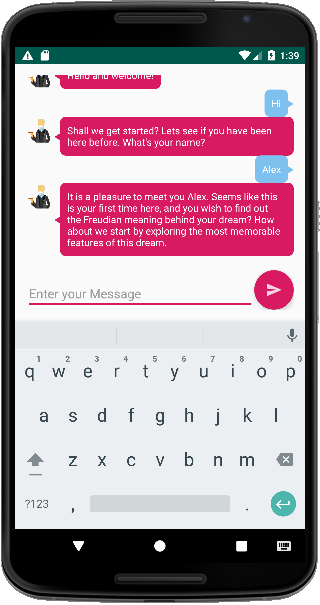
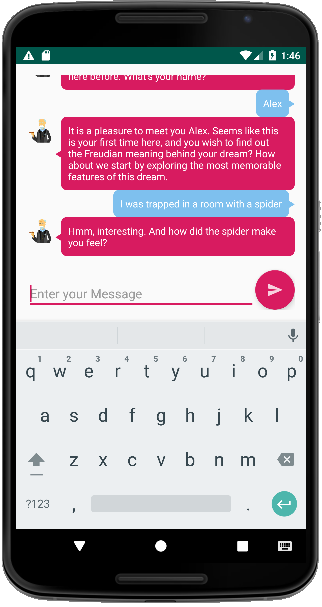
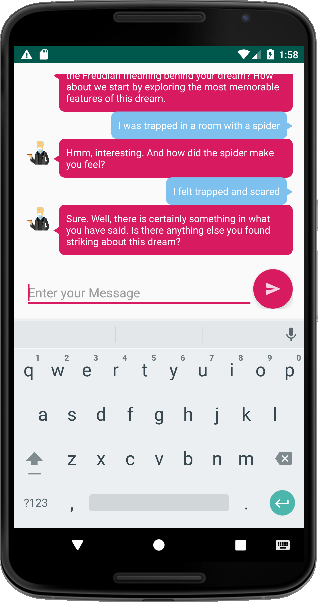
  

Figure 38 – Name Response

Figure 39 – Dream Description

Figure 40 – Dream Sentiment

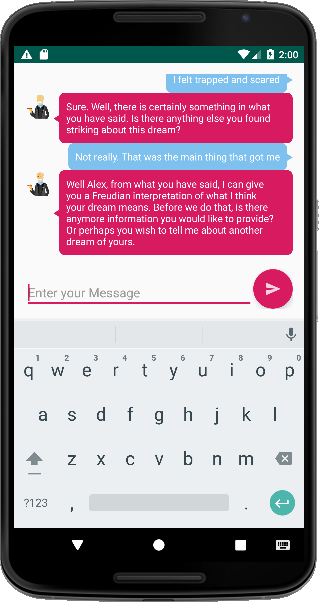
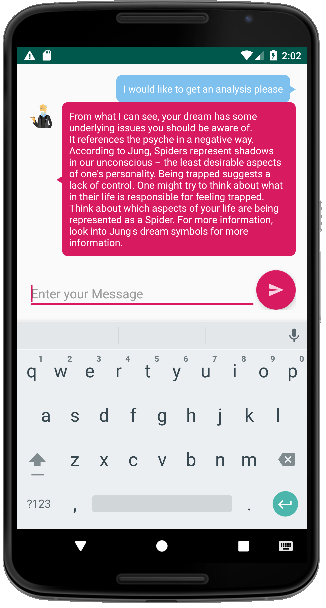
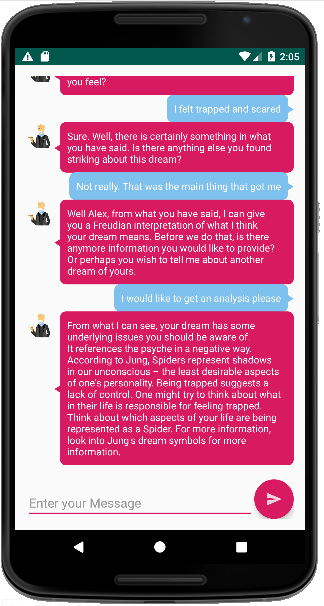
  

Figure 43 – Completion

Figure 42 – Analysis

Figure 41 – Post Options

Figures 35 – 43 show the simplest conversation required to complete a therapy session.

Figures 44 -51 show a possible means of comparing dreams.

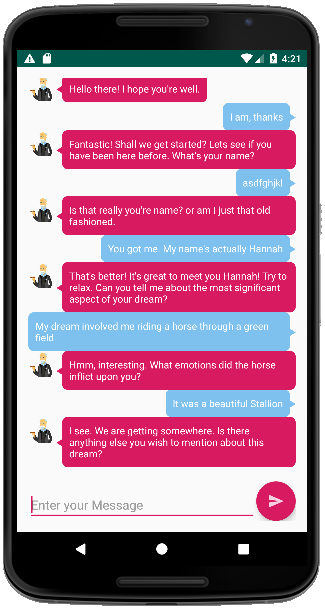
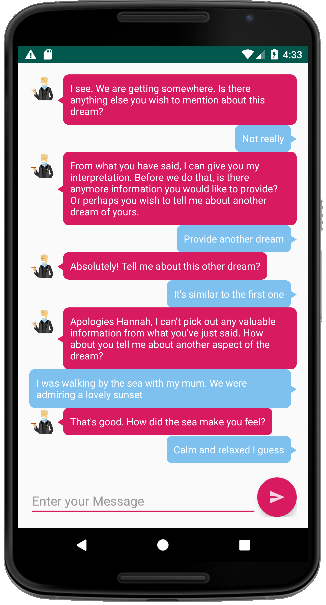
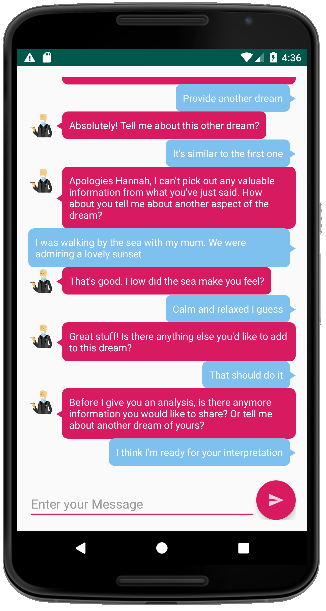
  

Figure 46 – 2nd Conversation

Part 3: Positive Dream

Figure 45 – 2nd Conversation

Part 2: No Dream

Figure 44 – 2nd Conversation

Part 1: No Name

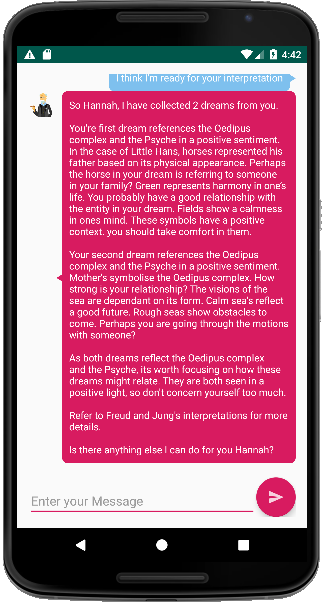


Figure 47 – 2nd Conversation

Part 4: Analysis

Figures 44 – 47 shows two dreams with the same symbols and the same sentiment. Figure 44 shows the consequence of an inappropriate name. Figure 45 shows an unrecognised dream.

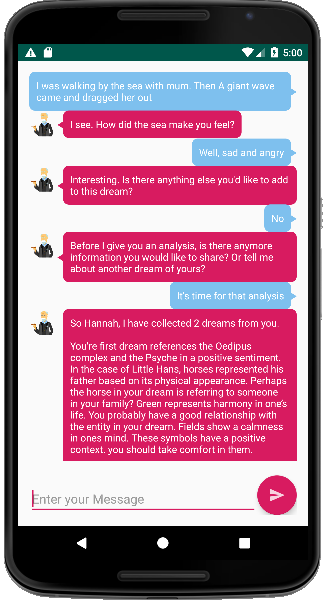
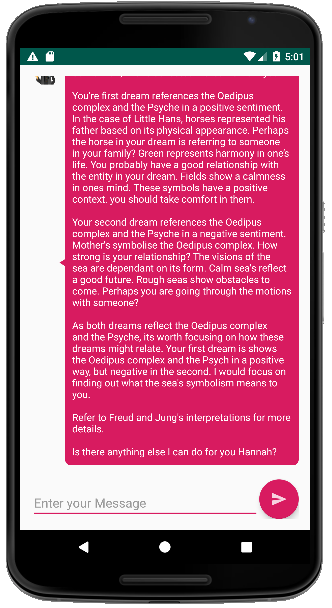
 

Figure 48 – 3rd Conversation

Part 1: Negative Dream

Figure 49 – 3rd Conversation

Part 2: Analysis

Figures 48 – 49 show two dreams with the same symbols but opposing sentiments.

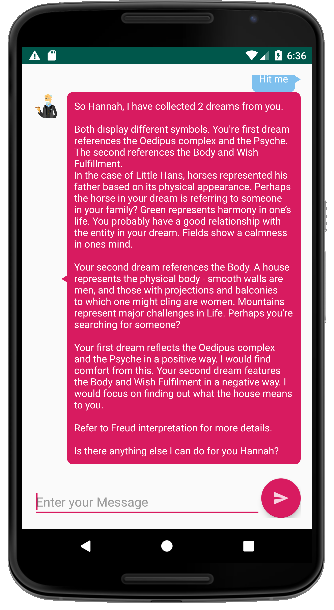
 

Figure 51 – 4th Conversation

Part 2: Analysis

Figure 50 – 4th Conversation

Part 1: Negative Dream

Figure 50 – 51 shows two dreams without matching symbols and different sentiments.

These prototypes show what an MVP should look like. The combination of pink and blue compliment a relaxed environment [33]. The interface is clean and simple. It is easily recognisable as a messenger and the icon of Freud shows the chatbot’s response.

With an idea of how to design Fralysis, the implementation process can begin.

# 7. Implementation

## 7.1 Programming Language

Python is the language used to develop Fralysis. PyCharm was chosen as the IDE. This decision was made based on Python’s Natural Language support. As Machine learning is very new, Python has become the best language supported in this field. Unfortunately, this meant sacrificing the GUI support provided in a language like Java.

## 7.2 Fralysis’ MVC Architecture

An object-orientated approach was implemented using Test Driven Development [34].

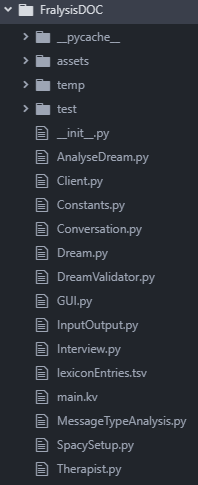
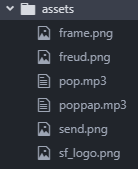
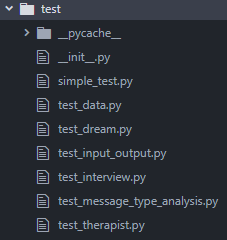
  

Figure 52 – Project Structure

Figure 54 – Tests

Figure 53 – Assets

Figure 52 shows the classes. This reflects Figure 32. Figure 53 shows the assets: a frame for the home page, pictures of Freud and sounds for button pressing. Figure 54 shows the tests described in section 8.

### 7.2.1 Model

The statistical model used to initialise the NLP pipeline was SpaCy’s ‘en\_core\_web\_sm’ model: ‘*English multi-task CNN trained on OntoNotes. Assigns context-specific token vectors, POS tags, dependency parse and named entities.’* [35]

As well as this, a Freudian lexicon was used to extend the pipeline to account for Freudian topics and ratings.

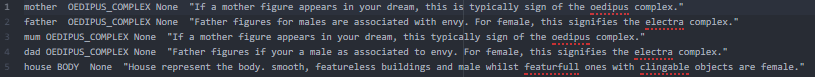


Figure 55 – lexiconEntries.tsv Snippet

Figure 55 shows a snippet of the ‘.tsv’ file used to populate the lexicon. It is based on Freudian interpretations [38]

The first column contains the direct object or compliment. This is followed by its Freudian symbol. The third column gives a rating. The final column gives the latent meaning for the direct object. A compliment will have column two and four set to none. The origins of these definitions can be found in Appendix 3.

Ideally, the ‘.tsv’ file should be replaced with an SQLite database. This would also serve the purpose of storing past client session data.

This feature can be added as a later update.

### 7.3.2 View

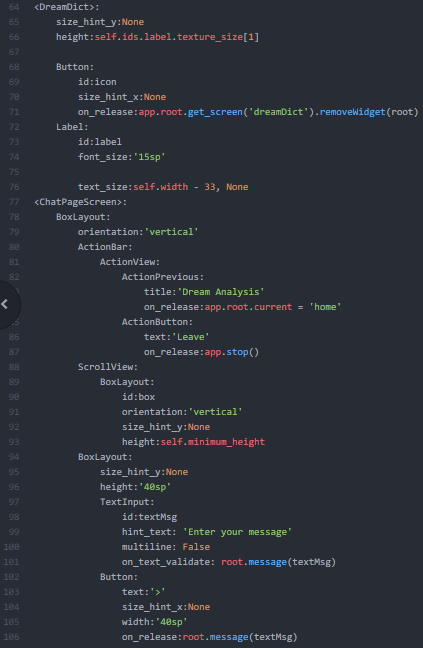
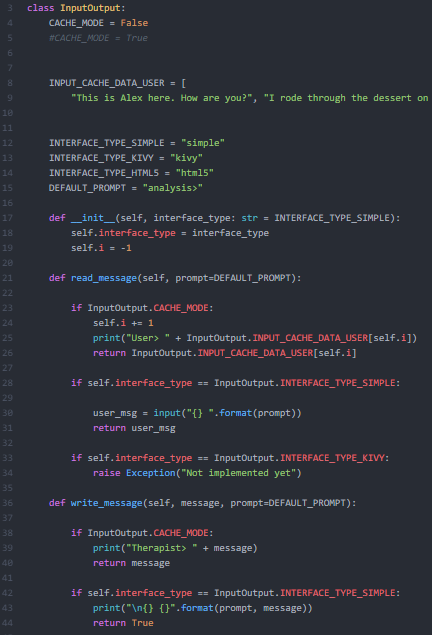
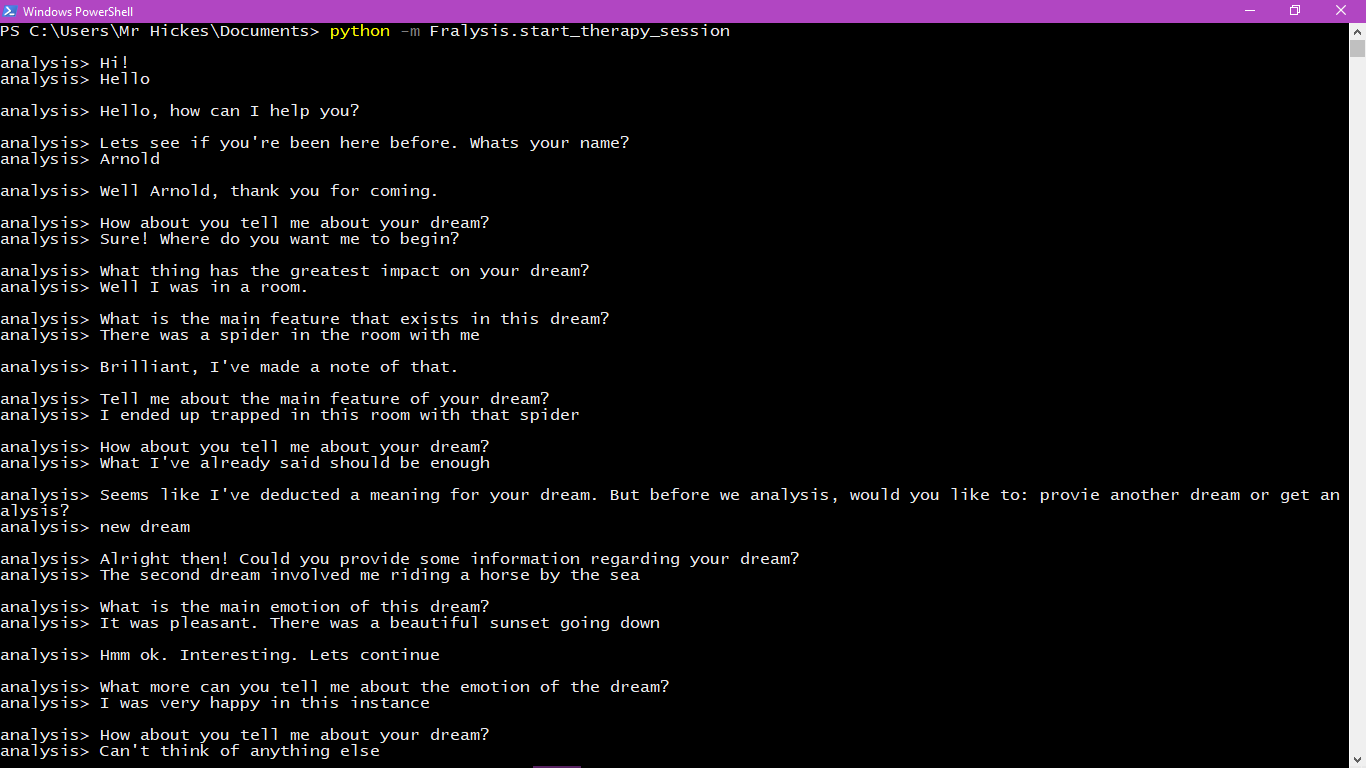
  

Figure 58 – InputOutput.py Snippet

Figure 57 – main.kv Snippet

Figure 56 – GUI.py Snippet

To keep the model separate from the view, Figure 56 contains the KIVY logic. Figure 57 contains the KIVY data. Figure 58 allows different views to initialise. Figure 56 Line 159 Shows the controller implemented as a thread. This maintains the MVC style.



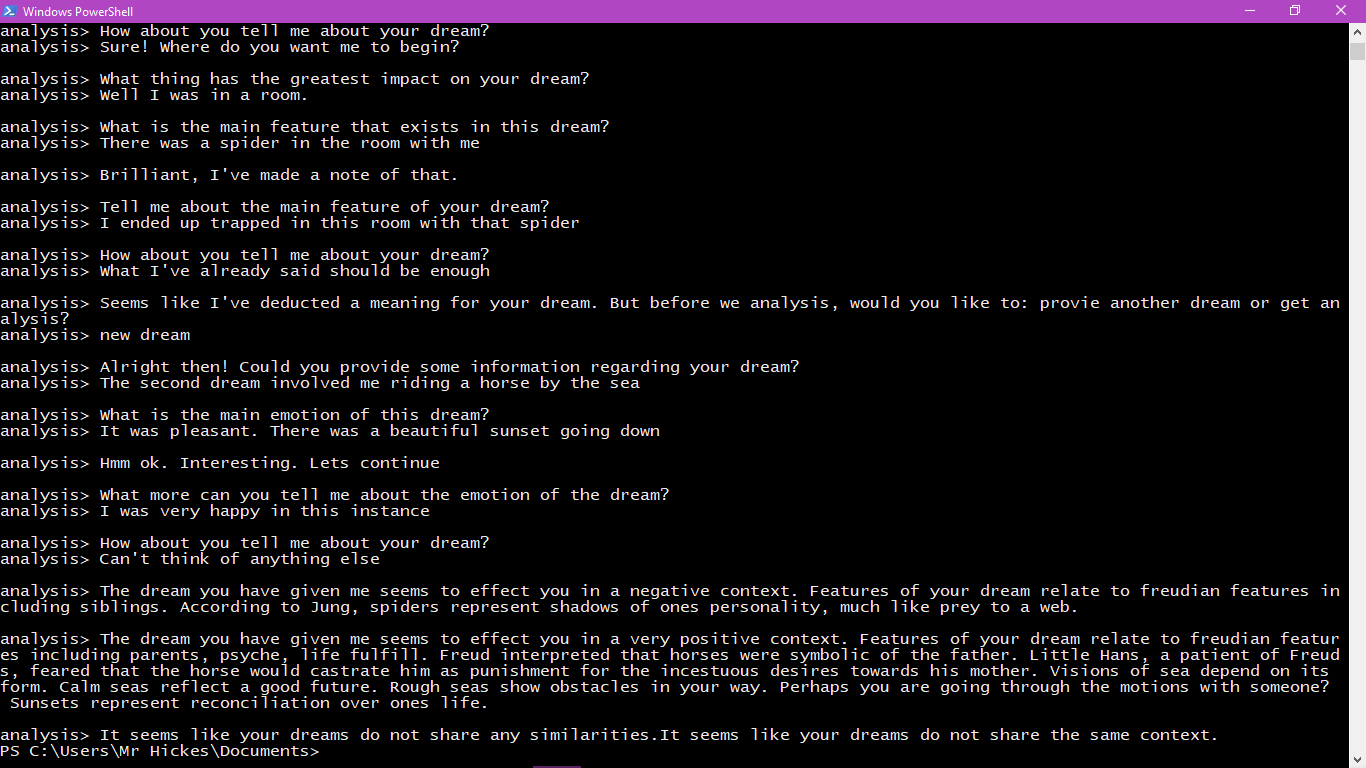


Figure 59 – Terminal View



Figure 60 – KIVY View

Buildozer [39] was used to export the app to an APK for Android. Buildozer provides a set of tools that enable Python/KIVY applications to be packaged for other platforms. Currently Buildozer can only be executed on Linux.

Issues occur while deploying the full ‘SpaCy’ version due to incompatible binaries.

04-24 18:11:08.607 14218 14238 I Python : ImportError: dlopen failed: “/data/data/org.fralysis.fralysis/files/app/\_Python\_bundle/site-packages/srsly/ujson/ujson.so” is 64-bit instead of 32-bit

04-24 18:11:08.607 14218 14238 I Python : Python for android ended.

Spacy has the following dependencies: spacy.thinc.srsly.ujson. As mentioned in 4.2.2.3, this is the neural model.

ujson contains ‘c’ components and c binary libraries. It seems like the 64-bit binary is incompatible with the target Android phone.

A Solution may involve using the web server approach mentioned in 4.2.2.5.

### 7.2.3 Controller

The bulk of the program involves simulating therapy. Most of this is can be found in 6.1.3.

## 7.3 Unique Feature

There were specific requirements Fralysis had to implement stated in 4.2.

### 7.3.1 Maintaining Therapist characteristics

The interview represents the therapist’s thought process.

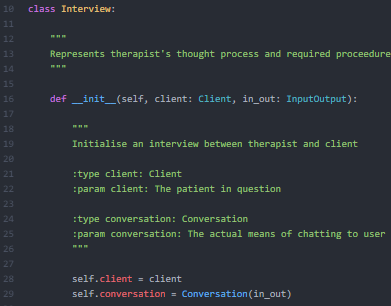
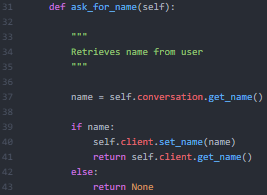
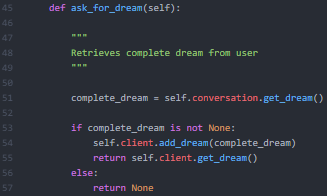
  

Figure 63 – ask\_for\_dream() Snippet

Figure 62 – ask\_for\_name() Snippet

Figure 61 – Interview: init() Snippet

The interview initiates one or more conversations and determines when the conversation cycle is complete – when the required information is collected.

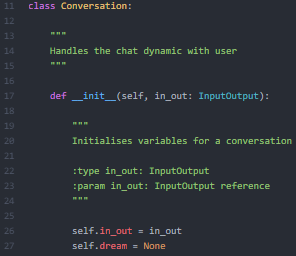


Figure 64 – Conversation: init() Snippet

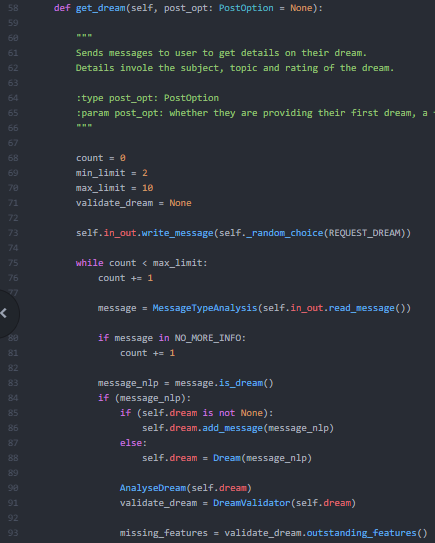
 

Figure 65 – get\_name() Snippet

Figure 66 – get\_dream() Snippet

The conversation requests information from the client in a human like fashion.

Figure 65 shows requests from standard to urgent, corresponding to Figure 67.

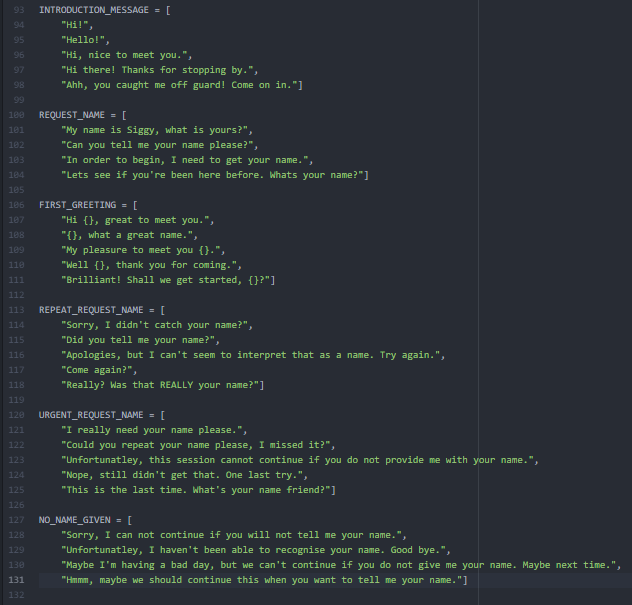


Figure 67 – Therapist Responses

Figure 67 shows lists with different phrases for requesting certain features. Responses are random. If the app is used multiple times, there should be variation in the chat dynamics.

Using this highly cohesive implementation, the chatbot feels real. The interview is direct. It does not deal with the complexities of users, but it is aware when the required information is collected.

### 7.3.2 Extracting User’s Name from Input

The most significant implementation was converting a string provided by the user into useful data. This begins with extracting the user’s name.

MessageTypeAnalysis has a method is\_name(). This iterates through the tokens in a sentence contained in a SpaCy Doc object until a token with a proper noun attribute is found.

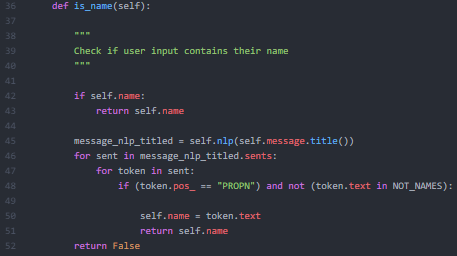


Figure 68 – MessageTypeAnalysis: is\_name() Snippet

This works well when given simple responses. When testing more convoluted phrases, words which are not nouns are interpreted as proper nouns based on the sentence structure. This has been a common issue presented on the SpaCy Github page [37].

This can be solved by loading SpaCy’s ‘en\_core\_web\_lg’ model [36] instead of ‘en\_core\_web\_sm’ [35]. This model only recognised true names as being proper nouns.

However, the large model exceeds 800+ Mb, as opposed to 29Mb. This led to responses going from within a second to over 10 seconds, which breached requirement F\_.

Therefore, the compromise has been to use the small model and assume the user keeps things simple. Most valid names are correctly identified.

Models have been designed to be configurable. Large models can be instantiated when using Fralysis on a desktop.

### 7.3.3 Extracting User’s Dream from Input

Recognising dreams requires extending the pipeline to accommodate for Freudian specific terms. Just as the pipeline attaches part of speech and dependency attributes to tokens, it is possible to add custom attributes.

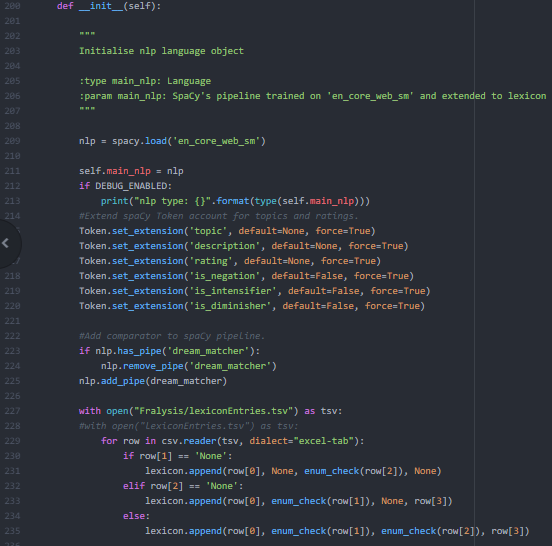


Figure 69 – SetupSpacy: init() Snippet

Figure 69 shows attributes required from tokens relating to Freud. For example, tokens like ‘mother’ and ‘father’ are distinctive features to Freud and relate to the Oedipus / Electra Complex.

As these ideas are exclusive to Freud, the default SpaCy model does not provide this data.

A lexicon is defined: ‘The vocabulary of a person, language, or branch of knowledge’.

This class is used to extend the default SpaCy model to account for Freud’s ideas.

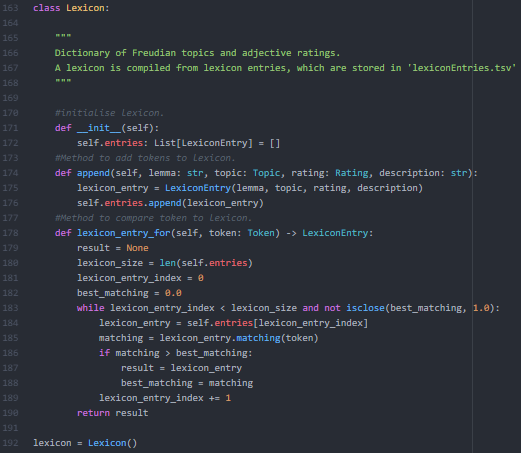


Figure 70 – SetupSpacy: Lexicon Snippet

Figure 70 shows a class representing a dictionary data structure containing lexicon entries.

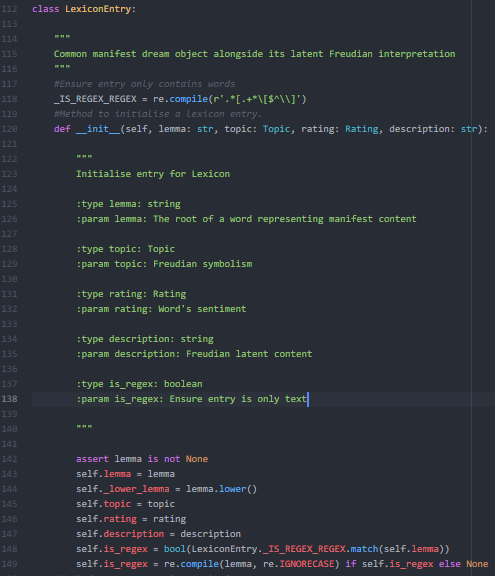


Figure 71 – SetupSpacy: LexiconEntry Snippet

Figure 71 shows a lexicon entry. It contains a lemma – the simplest form of a word.

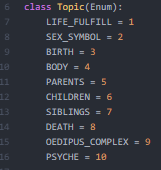
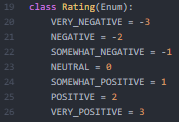
 

Figure 72 – Constants: Topic Snippet

List

Figure 73 – Constants: Rating Snippet

Snippet

List

Topics associate a manifest content with a symbol – ‘Oedipus Complex’, ‘Psyche’ and so on.

This allows multiple levels of analysis – instead of directly associating manifest content to Freudian interpretation. If the Oedipus Complex is present multiple times, this is fed back to the user.

A manifest content needs a sentiment. This makes it possible to identify whether the subject or topic is portrayed positively/negatively.

As seen from Figure 73, the Rating enum has seven values. All adjectives should have a rating attached.

This alone is only capable of analysing single adjectives. The idea of intensifiers, diminishers and negation must be introduced.

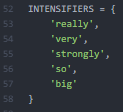
  

Figure 76 – Constants: Negation List

Figure 74 – Constants: Intensifiers List

Figure 75 – Constants: Diminishers List

Intensifiers increase the power of an adjective. Describing something as very good is more positive than just good. Fortunately, there are only a few common words that fit into this category, as shown in Figure 74. Common diminishers are given in Figure 75 and Negation in Figure 76.

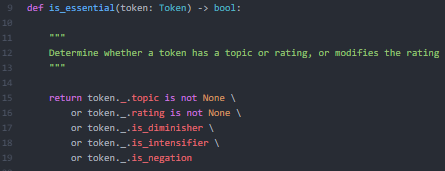


Figure 77 – AnalyseDream: is\_essential() Snippet

When getting the topic and sentiment of a sentence, tokens pass through a function called ‘is\_essential’ (Figure 77). If a rating token is identified, ‘combine\_rating’ (Figure 78) checks if there is anything modifying this token.

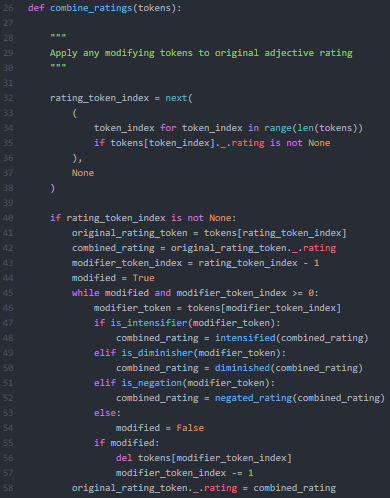


Figure 78 – AnalyseDream: combine\_rating() Snippet

Using the example ‘not very scary’, ‘scary’ is an adjective giving a very negative sentiment.

‘Combine\_rating’ looks to the left and finds an intensifier present – ‘very’. This will increase the value of the adjective by one. ‘Scary’ has a value of -3, so signum will add -1 to this, leaving a value of -4. This exceeds the rating scale, so an internal method ‘\_ranged\_rating’ keeps the value within the possible range defined by the very positive / negative limits (-3 to 3).

A further examination finds negation – ‘Not’. The term ‘scary’ is very negative, but something being ‘not scary’ does not make it very positive – It makes it somewhat positive. Figure 79 shows the negated map used to handle this condition.

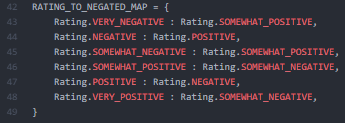


Figure 79 – Constants: RATING\_TO\_NEGATED\_MAP Snippet

This is a simplified model. It handles terms like ‘not good’.

Tokens with topics need latent descriptions (Figure 71). This resembles a dictionary that is referenced in the analysis.

Successfully implementing the Controller means the required data can be extracted from the user.

However, there are some vital issues to be illustrated. Freudian psychology is not a science: it is pseudo – science, so topics given to Direct Objects are based on previous results from Freudian analysis. Psychoanalysis requires extensive details: ‘*Freud undermined this naive theory of symbolic equivalences’* [6].

Dream analysis is a personal analysis, and Freud was ‘cautious’ when defining symbols: ‘*He did determine that there were some possibilities for universal symbols, such as poles, guns, and swords representing the penis.*’ [38]

Giving horses an association to the Oedipus Complex is fair in the case of Little Hans. However, horses can be interpreted indefinitely, and it is likely that a horse has been attributed to a completely different topic through a different psychoanalysis session.

To get a truly valid Freudian analysis of a dream, deeper machine learning is needed to replace the lexicon in Figure 70. By training the pipeline on Freud’s books/research, you could extensively question the user until the key dream features resemble a past Freudian case.

The issue with implementing this is time constraints. It would involve converting analogue corpuses to digital. Even if you did this, attitudes and daily influencers are so different now that it would be hard to find connections between modern dreams and dreams from the past due to technology and science, and life being so different.

It is worth re-iterating the true purpose of this project: provide a fun and experimental insight into Freudian psychology and dream analysis, with an aim to provide some history on some of Freud’s ideas.

The way Fralysis aims to deal with this issue is by researching key topics. As found here: *’According to Freud, the number of things represented by symbols in dreams is not great: The human body, parents, children, siblings, birth, death, nakedness, and a few others*’ [38]. These are included in the lexicon (Figure 70).

The implementation of Fralysis accurately reflects the design. An MVC architecture is successfully implemented, allowing interchangeable models and views. A detailed overview is given in section 9.

The MVC components resulted from Test Driven Development.

# 8. Testing

## 8.1 Unit Testing

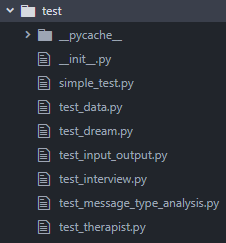
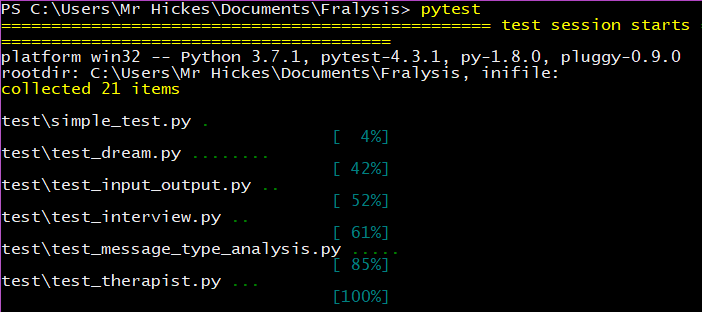


Figure 80 – Unit Test Structure

Fralysis was a product of test-driven development [34]. Each class had specific purposes. Tests were written for individual functionalities, and the class was implemented until the tests passed.

## 8.2 Module testing



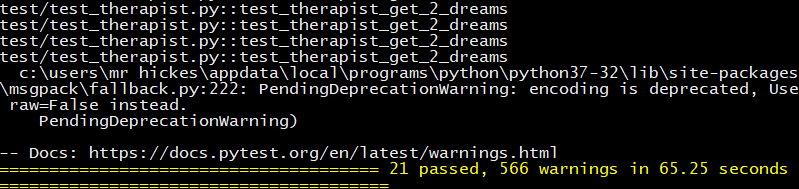


Figure 81 – PyTest Snippet

As well as Individual tests, cached data was used to automate conversations (Figure 81). These confirm the flow logic in 6.1.1.

## 8.3 User testing

This involved giving the application to friends and requesting feedback from them. They all complied to the Ethical Compliance form found in the Appendix 1.

Five friends were given the application to play with and answered five questions stated in Figure 82. Answers were quantitative: 1 = strongly disagree and 5 = strongly agree.

|  |  |
| --- | --- |
| Question | Average response |
| My messages lead to an analysis being provided | 5/5 |
| I was able to understand the analysis | 4/5 |
| The analysis taught me something about Freud | 3/5 |
| The chatbot felt like a human | 4/5 |
| The chatbot gave the impression of being a therapist | 3/5 |
| The conversation felt natural | 4/5 |

Figure 82 – User testing results

Figure 81 shows all five participants ended up with an analysis at the end of the session. This proves the flowchart in 7.1.1 was designed effectively.

From the analysis, most participants could decipher the meaning of the dream. This was given with a reference to appropriate Freudian history.

Most participants thought the chatbot had human-like characteristics. When asked why, participants replied: ‘It felt more real than Siri or Alexa in the way it answered questions’. Whether it gave a therapist impression was mixed. This was due to enforcing the requirement of: experiment and fun. Most participants agreed that it felt like an ‘eccentric therapist’. Given the nature of Freud, you could class this as somewhat positive.

Participants agreed it felt like a natural conversation.

# 9. Evaluation

Below provides an overview of this project. Technical descriptions on the specifics mentioned below can be found in the implementation Fralysis can be found in section 7.

## 9.1 Successes

The flow of the therapy session was successfully implemented. Github provided a repository and version control for the project [1]. This also includes code documentation in HTML format generated using Sphinx [40].

A valid Freudian data model has been comprised from Freud’s ‘Introduction to psychoanalysis’ [4].

The KIVY view accomplished its purpose of providing a GUI which enabled a user to interact with the chatbot. It allowed for successful user testing. Fralysis was successfully built for Windows 10 and Ubuntu.

An explanation of this can be due to an opportunity I had whilst developing this app. Whilst working for a games research company, I realised their process of evaluating qualitative data provided by players was very similar to my approach – find relevant references to the game, tag this and attach the players thoughts on it.

By substituting the Freudian lexicon for a game specific dictionary, I was able to test my feature extractor against my colleague’s analysis – who are professional researchers.

This led to a focus on the NLP aspects of Fralysis. Testing this capability required a spoof data cache to enable fully automated testing of any required user dialogue.

Due to confidentiality, I can’t provide details of data analysed, but it worked remarkably well in this context.

## 9.2 Limitations

These can be found at the end of each subsection in section 9.

The model should have been implemented as an SQLite database. However, the time required to do this was longer than expected. The tsv. file suffices in providing a lexicon data file.

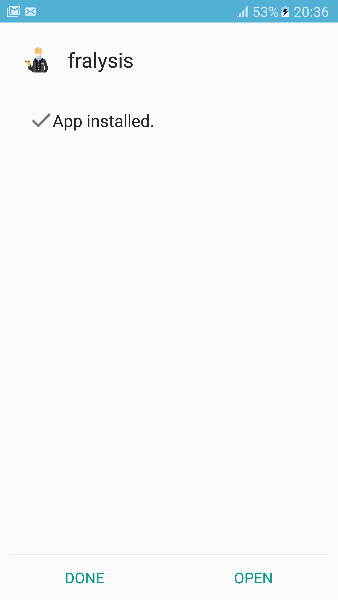
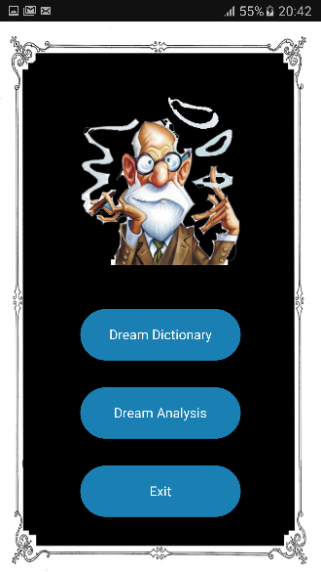
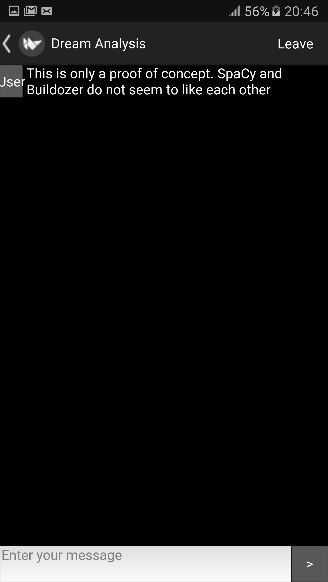
   

Figure 83 – Packaging Fralysis with Buildozer and installing on Samsung s7

Although Fralysis could be installed on an Android phone, the SpaCy libraries had to be removed. This meant it was installed without a model, and no functionality.

Given the time period to research and develop this project, it has been created in a limited capacity. Although there are features which took longer than expected, the features implemented are supported with thorough documentation and testing.

## 9.3 Overall

Overall, I have conducted a project revolving around Freudian dream analysis and considered the professional considerations whilst doing so.

As this is a very abstract, non-scientific subject, background research was required to understand what this project would exactly involve.

Types of dreams had to be classified and understanding why someone might desire to get them analysed was important.

Freud’s ideas on dream analysis were overviewed. This provided the criteria required from a Freudian analysis. This looks at manifest and latent content as well as dream symbols.

The approach used to extract these features was Natural Language Processing. This looks at what nlp pipeline means and considers sentence structures that need to be analysed to achieve this. This was successfully achieved using Explosion AI’s SpaCy.

For the front end, possible interfaces were evaluated. As the implementation involves a chatbot, a standard messenger interface was used. This was successfully achieved using KIVY but did not couple with SpaCy. As a proof of concept, the GUI for the application, excluding the ‘SpaCy’ libraries has been successfully built and deployed on Android.

Fralysis was implemented and given to users to test, who provided helpful feedback.

The next steps for development will be to resolve issues between SpaCy and KIVY to deploy a working model on a mobile. I would also attempt to either expand the lexicon or try a more machine learning approach.

# References

[01] Lauren, L. (2014, 01 21). *Four Types of Dreams*. Retrieved from HuffPost: https://www.huffingtonpost.ca/linda-lauren/dream-interpretation\_b\_4639124.html?guccounter=1&guce\_referrer\_us=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce\_referrer\_cs=AEDTzZ5mGmBigU1L\_Mj65A

[02] Cobarrubias, S. (2003). *Dream Analysis: Definition and Example*. Retrieved from Study.com: https://study.com/academy/lesson/dream-analysis-definition-examples-quiz.html

[03] Freud, S. (1899). *The Interpretation of Dreams.* Schloss BelleVue: Franz Deuticke, Leipzig & Vienna.

[04] Freud, S. (1920). *A General Introduction to Psychoanalysis.* Schloss BelleVue: Liveright.

[05] Dream Content. (2019, 04 19). *Wikipedia*. Retrieved from Content (Freudian dream analysis): https://en.wikipedia.org/wiki/Content\_(Freudian\_dream\_analysis)

[06] Freud Museum. (2018). *Freud’s Method for Interpreting Dreams*. Retrieved from Freud Mueseum London: https://www.freud.org.uk/learn/discover-psychoanalysis/the-interpretation-of-dreams/freuds-method-for-interpreting-dreams/

[07] Turing, A. (1950). *Computing Machinery and Intelligence.* Oxford: Oxford University Press.

[08] Avalverde, D. (n.d.). *A Brief History of Chatbots*. Retrieved from Perception, Control, Cognition: https://pcc.cs.byu.edu/2018/03/26/a-brief-history-of-chatbots/

[09] Weizenbaum, J. (1966, 08 21). *ELIZA*. Retrieved from Eclectic Energies: https://www.eclecticenergies.com/ego/eliza; https://www.masswerk.at/elizabot/

[10] Colby, K. (1973, 07 13). *PARRY*. Retrieved from CMU Artificial Intelligence Repository: http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/classics/parry/0.html; https://phrasee.co/parry-the-a-i-chatterbot-from-1972/

[11] Wallace, R. (1995, 06). *A.L.I.C.E*. Retrieved from Pandora: https://www.pandorabots.com/pandora/talk?botid=b8d616e35e36e881

[12] Brigham Young University. (2018, 04 12). *Programming socialbots: BYU competes in Alexa Prize Challenge*. Retrieved from Youtube: https://www.youtube.com/watch?v=9bSVpv3aoww

[13] Mody, P. (2018, 03 06). *What is a Training Data Set & Test Data Set*. Retrieved from Quora: https://www.quora.com/What-is-a-training-data-set-test-data-set-in-machine-learning-What-are-the-rules-for-selecting-them

[14] Todorović, A. (2015). *Has The Turing Test Been Passed?* Retrieved from Github: http://isturingtestpassed.github.io/

[15] Jones, K. S. (2001). *Natural language processing: a historical review.* Cambridge: University of Cambridge.

[16] Anbananthen, K. S. (2017). *Comparison of Stochastic and Rule-Based POS Tagging on Malay Online Text.* Melaka: Malaysia Multimedia University.

[17] Kravitz, D. (2017, 02 28). *Business career of Donald Trump*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Business\_career\_of\_Donald\_Trump

[18] Stevens, S. (2009, 08). *Sentence Patterns*. Retrieved from University of Houston: https://www.uhv.edu/student-success-center/resources/q-z/sentence-patterns/

[19] Freud, S. (1909). *Little Hans.* Schloss BelleVue: Pelican Freud Library. Retrieved from SimplyPsychology: https://www.simplypsychology.org/little-hans.html

[20] Neufeld, B. (2018, 05 22). *Range.io*. Retrieved from Chatbots: An Introduction to Conversational UI: https://rangle.io/blog/chatbots-an-introduction-to-conversational-ui/; https://www.youtube.com/watch?v=SvV57fuL\_M0&t=787s

[21] Jones, J. (2018, 15 06). *Free Association*. Retrieved from GoodTherapy: https://www.goodtherapy.org/blog/psychpedia/free-association-in-therapy

[22] Hawkins, J. (2005). The Philosophy of the Handheld. In D. Stone, *User Interface Design and Evaluation* (p. 379). UK: Open University.

[23] iDream App. (2018). Dream App. By iDream, https://idre.am/.

[24] DREAM-e. (2012). Dream App. By Techsophics, http://technosophics.com/home/dream-e/.

[25] MorpheusDreams. (2012). Dream App. By Morpheus Dreams, http://morpheusdreamsapp.com/.

[26] Ling, J. (2014, 04 07). *Java or Python for Natural Language Processing.* Retrieved from stackoverflow: https://stackoverflow.com/questions/22904025/java-or-Python-for-natural-language-processing

[27] Honnibal, M. (2014). *SpaCy.* Retrieved from ExplosionAI: https://spacy.io/

[28] Honnibal, M. (2016). *Tokenizer.* Retrieved from SpaCy Documents: https://spacy.io/api/tokenizer

[29] Honnibal, M. (2016). *Tagger.* Retrieved from SpaCy Documents: https://spacy.io/api/tagger

[30] Honnibal, M. (2016). *Model*. Retrieved from SpaCy Documentation: https://spacy.io/models

[31] Autin, V. (2011). *KIVY.* Retrieved from https://kivy.org/#home

[32] Freud, S. (1880). *Anna O.* Schloss BelleVue: Pelican Freud Library.

[33] Decor, D. (2005). *Color Combinations*. Retrieved from Bright Side: https://brightside.me/article/the-ultimate-color-combinations-cheat-sheet-92405/

[34] Beck, K. (2002). *Test Driven Development.* Oregon: Addison-Wesley Professional.

[35] Honnibal, M. (2019, 03 17). *SpaCy Model: en\_core\_web\_sm-2.1.0*. Retrieved from Github: https://github.com/explosion/spacy-models/releases/tag/en\_core\_web\_sm-2.1.0

[36] Honnibal, M. (2019, 03 17). *SpaCy Model: en\_core\_web\_lg-2.1.0*. Retrieved from Github: https://github.com/explosion/spacy-models/releases/tag/en\_core\_web\_lg-2.1.0

[37] Honnibal, M. (2018, 06 7). *Multi-word Entity*. Retrieved from Github: https://github.com/explosion/spaCy/issues/2421

[38] McAndrew, F. T. (2018, 01 01). *The Freudian Symbolism in your Dreams*. Retrieved from SimplyPsychology: https://www.psychologytoday.com/us/blog/out-the-ooze/201801/the-freudian-symbolism-in-your-dreams

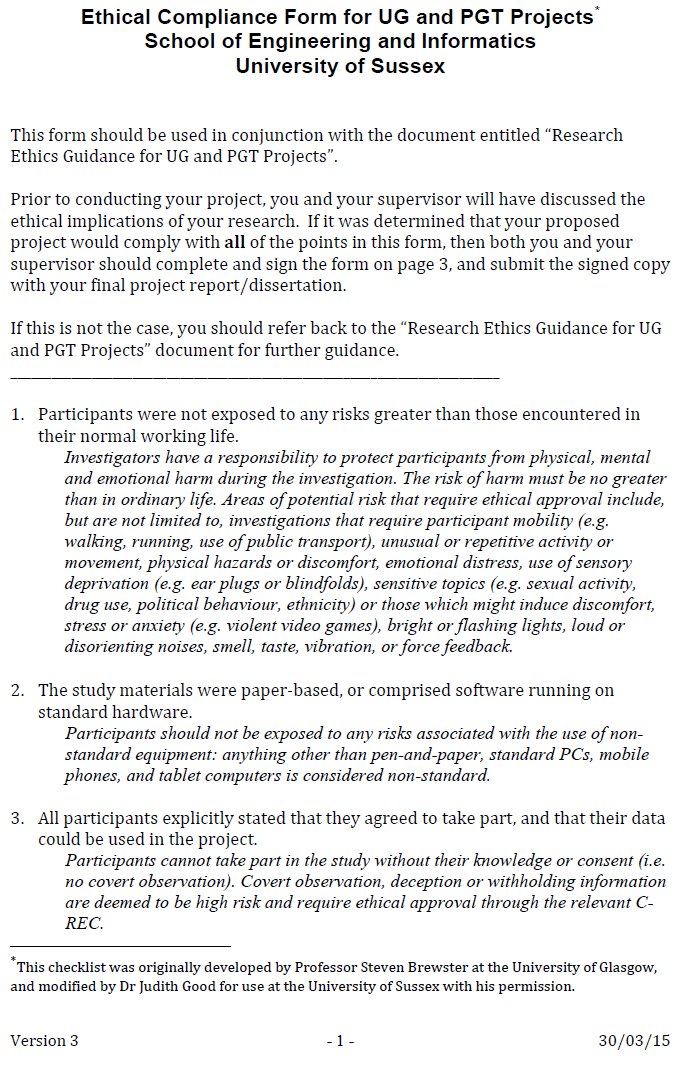
[39] Virbel, M. (2014). *Buildozer Documentation*. Retrieved from Buildozer: https://buildozer.readthedocs.io/en/latest/

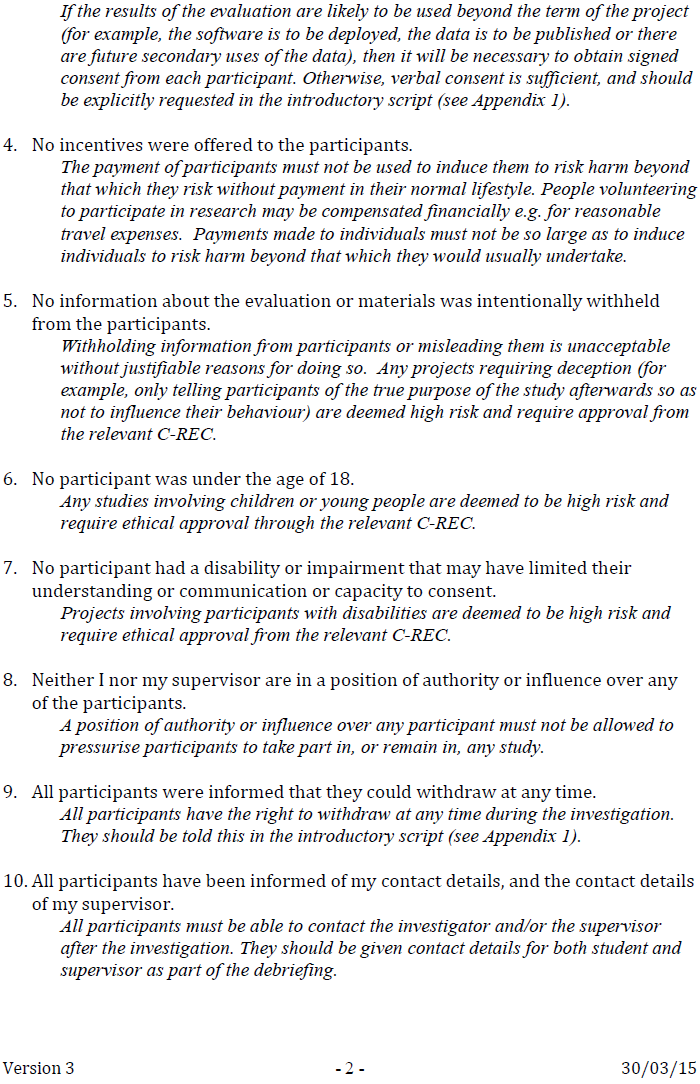
[40] Brandl, G. (2007). *Sphinx Documentation*. Retrieved from Sphinx: http://www.sphinx-doc.org/en/master/

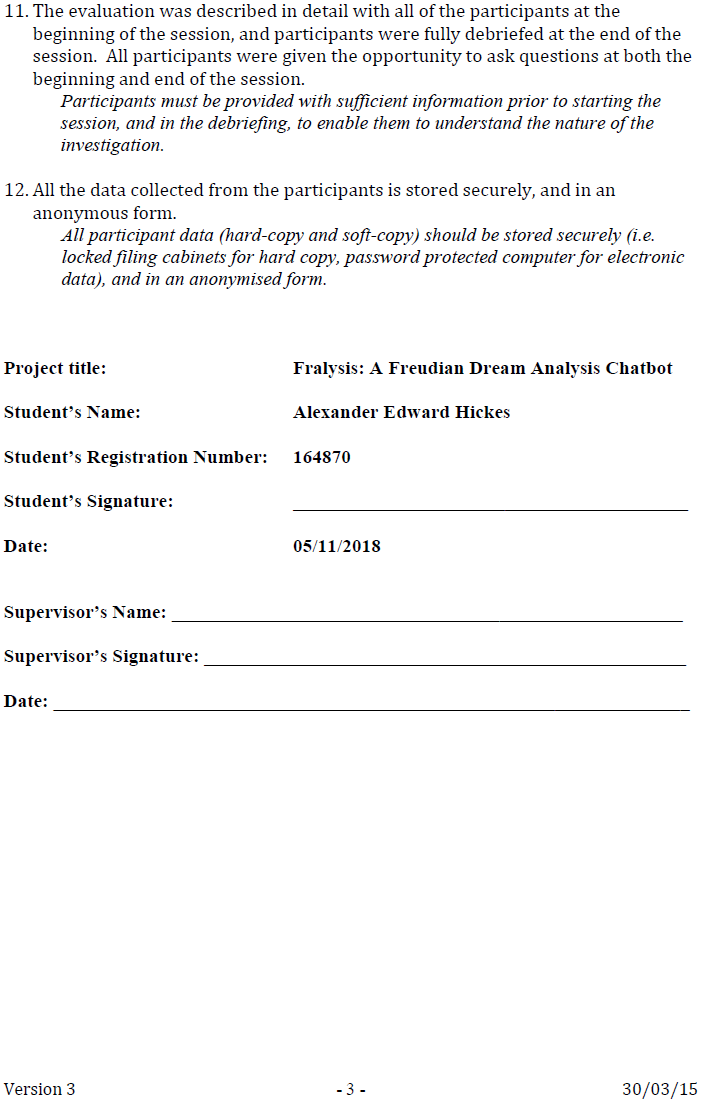
[41] Hickes, A. (2019, 04 29). *Repository for Project*. Retrieved from Github: https://github.com/alex-hickes/dream\_analysis.git

# Appendix

## Appendix 1: Ethical Consent Form







## Appendix 2: Project Plan



## Appendix 3: Freudian Symbols

|  |  |  |
| --- | --- | --- |
| Symbol | Freudian Meaning | Origin |
| Human Body | The person is often represented in the form of a house | <https://www.psychologytoday.com/>  gb/blog/out-the-ooze/201801/  the-freudian-symbolism-in-your-dreams  <https://www.bartleby.com/>  283/10.html |
| Oedipus (Electra) Complex | describe a child’s feelings of desire for his or her opposite-sex parent and jealousy and anger toward his or her same-sex parent |
| Parents | Parents appear in dreams as kings, queens, or other highly respected persons |
| Children | children are symbolized as vermin |
| Siblings | siblings are symbolized as small animals |
| Birth | Birth is almost always represented by some reference to water. |
| Death | Death is replaced in dreams by taking a journey |
| Nakedness | nakedness is symbolized by clothing, especially uniforms |
| Sex Symbol | the great majority of symbols in dreams are sex symbols |

## Appendix 4: Freudian Interpretations

|  |  |  |
| --- | --- | --- |
| Word | Freudian interpretation | Origin |
| Mother | Parents appear in dreams as kings, queens, or other highly respected persons | <https://www.bartleby.com/>  283/10.html  <https://www.psychologytoday.com/>  gb/blog/out-the-ooze/201801/  the-freudian-symbolism-  in-your-dreams |
| Father |
| House | The person as a whole is often represented in the form of a house – houses with smooth walls are men, and those with projections and balconies to which one might cling are women |
| King | Highly respected figures represent parents |
| queen |
| Water | Represents birth |
| Seed |
| Journey | Journeys represent death |
| Naked | Represents the body |
| Three | Three is a symbolic substitute for the entire male genital |
| Bath | Represent nakedness |
| Tree | The penis alone is represented by long and upright objects such as sticks, umbrellas, poles, trees |
| Fountain | male genitalia are objects out of which water flows, such as faucets and fountains |
| Waterfall |
| Sword | objects that can penetrate the body and cause injury – think of knives, daggers, lances, swords, and firearms (especially revolvers) |
| Gun |
|  |
| Ship | female genitals are symbolically represented by objects which enclose a space capable of being filled by something |
| Flying | penis raises itself against the force of gravity results in symbolic representation by balloons, airplanes, missiles, and rockets |
| Snake | less obvious male symbols include reptiles, especially snakes, and fish, hats, and coats. |
| Fish |
| Mouth | female genitals are represented by objects which enclose a space capable of being filled by something |
| Church |
| Bottle |
| Box |
| Sweets | sweets frequently stand in for sexual delight |
| Rocket | penis raises itself against the force of gravity results in symbolic representation by balloons, airplanes, missiles, and rockets |
| garden | garden a frequent symbol of the female genitals | <https://www.bartleby.com/>  283/10.html |
| Mountain | Mountains and cliffs are symbols of the male organ |
| Fruit | Fruit does not stand for the child, but for the breasts |
| Train | Death is replaced in the dream by taking a journey, riding in a train |
| Plane | raise itself against the force of gravity, one of the phenomena of erection, leads to symbolic representations by balloons, aeroplanes, and more recently, Zeppelins |
| snake | less comprehensible male sex-symbols belong certain reptiles and fish, notably the famous symbol of the snake |
| Flying | flying, often so beautiful, and which we all have had, must be interpreted as dreams of general sexual excitement, as erection dreams |
| Car | Generally, these types of dreams occur when you are feeling insecure. A car crash in a dream can indicate a lack of confidence. Maybe you are losing something that you care about, including a job, relationship, home or other aspects of your life. | <https://www.auntyflo.com/>  dream-dictionary/cars |
| Sunset | feeling sadness over your circumstances, which the sunset might symbolize. But the sun going down might also mean that the end of your trials is as near as the next coming day. | <https://exemplore.com/>  dreams/  How-to-Interpret-the-Sun-  as-a-Dream-Symbol |
| Forest | According to Freud's theory of psychoanalysis, the woods can be interpreted as a conducive environment for romance. In your waking life, things may be falling into place for your hidden desires and suppressed romantic energies to come out and play. | <https://www.dreamlookup.com/>  index.php/search/level1/forest/ |
| Horse | Horses were particularly suitable father-symbols because of their large penises | <https://www.simplypsychology.org/>  little-hans.html |
| Sea | Freud argues that those experiencing an oceanic feeling as an adult are experiencing a preserved primitive ego-feeling | <https://en.wikipedia.org/>  wiki/Oceanic\_feeling |
| Spider | spider in dreams is a symbol of the mother, but of the phallic mother, of whom we are afraid; so that the fear of spiders expresses dread of mother-incest and horror of the female genitals | <http://www.freudfile.org/>  psychoanalysis/  symbols.html |
| Island | area of your unconsciousness hidden away from the world | <https://dream-meaning.net/>  place/island-dream-interpretation/ |
| Desert | lack of nourishing ideas and energy in your life | <http://dreamstop.com/>  desert-dream-symbol/ |
| White | New experience and feelings | <http://www.dreamgate.com/>  dream/hoss/ |
| Black | The unconscious realm. Moving into darkness = suppression, “death of the ego” (first stage of transformation). Beautiful shiny black = a positive view of the unconscious from which a new self and new potentials emerge |
| Blue | eel tranquil, peaceful and content |
| Red | feel intense, vital or animated |
| Yellow | joy and optimism |
| Green | establish myself, my self-esteem, my independence |
| Silver | feminine qualities and also how we interact with others |
| Gold | heart, compassion, persistence, and patience |
| Orange | expand my interests and develop new activities |
| Pink | feel romantic or loving toward someone or something |
| Purple | feeling erotic and intimate |
|  | After all, as Freud himself once said, “sometimes a cigar is just a cigar.” | |

## Appendix 5: Sample Dreams

|  |  |  |
| --- | --- | --- |
| Dream | Freud’s Interpretation | Origin |
| “On July 13, 1910, toward morning, I dreamed that I was bicycling down a street in Tübingen, when a nice brown Dachshund car tore after me and caught me by the heel. Two elderly ladies are sitting opposite me and watching me with grins on their faces. Then I wake up and, as so often happens to me, the whole dream becomes perfectly clear to me in this moment of transition to the waking state.” | The most pleasant means might have been the Dachshund … “I lately fell in love with a girl, just from seeing her on the street, but had no means of becoming acquainted with her. The most pleasant means might have been the Dachshund, since I am a great lover of animals, and also felt that the girl was in sympathy with this characteristic.” … Perhaps the elderly ladies who simpered at him took the place of the girl | [www.bartleby.com/](http://www.bartleby.com/)  283/12.html |
| “Father is dead, but has been exhumed and looks badly. He goes on living, and the dreamer does everything to prevent him from noticing that fact.” | when he came back from his father’s funeral, one of his teeth began to ache ... the tooth is not taken out, but something that has died off is taken out of it |
| “She is going through the hall of her house and strikes her head against the low-hanging chandelier, so that her head bleeds.” | Thus the head here stands for the other part of the body. … results from its collision with the male organ … her belief that menstrual bleeding results from sexual intercourse with a man |
| “She sees a deep hole in the vineyard which she knows was made by pulling out a tree.” | The dream deals with another bit of the infantile sex theory, namely, with the belief that girls originally had the same genitals as boys and that the later conformation resulted from castration (pulling out of a tree) |
| “She is standing in front of the drawer of her writing table, with which she is so familiar that she knows immediately if anybody has been through it.” | The writing-table drawer, like every drawer, chest, or box, stands for the female genital … signs of sexual intercourse (and, as she thinks, even of any contact at all) and she has long been afraid of such a conviction. I believe that the accent in all these dreams is to be laid upon the idea of knowing. She is reminded of the time of her childish sexual investigations, the results of which made her quite proud at the time. |
| “An officer with a red cap follows her on the street. She flees from him, runs up the staircase, and he follows after her. Breathlessly she reaches her apartment and slams and locks the door behind her. He remains outside and as she looks through a peephole she sees him sitting outside on a bench and weeping.” | the pursuit by an officer with a red cap, and the breathless stair climbing, the representation of the sexual act. … the discharge of the semen is also indicated. … in psychoanalysis it is always maintained that all dreams have a sexual meaning. … But that the markedly distorted dreams preponderantly—though again not exclusively—give expression to sex wishes, is a fact you may certainly keep in mind as one of the results of psychoanalytical research |
| “Then someone broke into her house and she called in fright for a watchman. But the latter had gone companionably into a church together with two ‘beauties.’ … This was overgrown on both sides with grass and underbrush that kept getting thicker and that became a regular forest on the crest of the hill.” | The male genital is represented by a trinity of persons, the female by a landscape with a chapel, hill and forest. Again you encounter steps as the symbol of the sexual act. That which is called a hill in the dream has the same name in anatomy, namely, mons veneris, the mount of Venus |
| “He is traveling in a train. The train stops in an open field. He thinks it means that there is going to be an accident, that he must save himself, and he goes through all the compartments of the train and strikes dead everyone whom he meets, conductors, engine drivers, etc.” | The insane man murdered his fellow passenger. … saw the girl whom he had expected to marry but whom he had left because she had given him cause for jealousy … he would have to strike dead all the persons who stood in his way. … I can assume with certainty that the wish to be as crazy as that nevertheless exists in him |

## Appendix 6: Log

|  |
| --- |
| Autumn Term – 1st Meeting |
| Meeting Type:   * Project topic   Meeting Date:   * 02/10/18   Start (Time):   * 14:00   End (Time):   * 14:30   Attendance:   * Supervisor: Chris Thornton * Alex Hickes   Progress:   * Introduce project aims * Introduce project activities * Discuss project topics   Actions:   * Decide core project topic * Email details * Make project plan * Identify core features * Conceptualise unique features * Background research * Identify similar products * Understand app development process   Next Meeting:   * 12/10/18 - 13:30 * Location: Chris’ office |
| Autumn Term – 2nd Meeting |
| Meeting Type:   * Project plan and proposal   Meeting Date:   * 12/10/18   Start (Time):   * 14:00   End (Time):   * 14:30   Attendance:   * Supervisor: Chris Thornton * Alex Hickes   Progress:   * Discuss project plan and receive advice * Discuss proposal and receive advice * Discuss app’s database (SQLite, JSON) * Discuss app’s logical capabilities (NLP, Chatbot) * Discuss app’s interface (Messenger)   Actions:   * Project plan modifications * Proposal modifications * Email updated plan and proposal * Analyse similar products * Further unique feature considerations   Next Meeting:   * 19/10/18 - 13:00 * Location: Chichester Lab 1 |

|  |
| --- |
| Autumn Term – 3rd Meeting |
| Meeting Type:   * Interim Report   Meeting Date:   * 19/10/18   Start (Time):   * 13:00   End (Time):   * 13:10   Attendance:   * Supervisor: Chris Thornton * Alex Hickes   Progress:   * Discuss interim report and receive advice   Actions:   * Separate required knowledge from similar technology research * Provide three existing solutions and compare   Next Meeting:   * 20/02/19 - 14:00 * Location: Chris’ Office |

|  |
| --- |
| Spring Term – 4th Meeting |
| Meeting Type:   * App progress   Meeting Date:   * 20/02/19   Start (Time):   * 14:05   End (Time):   * 14:25   Attendance:   * Supervisor: Chris Thornton * Alex Hickes   Progress:   * Discuss current Freudian model * Discuss chatbot dynamics * Discuss chatbot interface   Actions:   * Implement dream comparison * Conceptualise user’s previous dream history retrieval * Design app homepage   Next Meeting:   * 06/03/19 - 13:00 * Location: Chichester Lab 1 |

|  |
| --- |
| Spring Term – 5th Meeting |
| Meeting Type:   * Further app progress and poster   Meeting Date:   * 06/03/19   Start (Time):   * 13:00   End (Time):   * 13:30   Attendance:   * Supervisor: Chris Thornton * Alex Hickes   Progress:   * Discuss Freudian model modifications and interpretation origins * Discuss Interface implementation * Discuss poster concepts   Actions:   * Create graphics for interface * Create poster   Next Meeting:   * 30/04/19 - 15:00 * Location: Chris’ Office |

|  |
| --- |
| Spring Term – 6th Meeting |
| Meeting Type:   * Final report   Meeting Date:   * 30/04/19   Start (Time):   * 15:00   End (Time):   * 15:30   Attendance:   * Supervisor: Chris Thornton * Alex Hickes   Progress:   * Discuss key report sections * Discuss app compromises * Discuss loss of right-hand impairment and extensions   Actions:   * Illustrate ideal prototypes and proof of concepts * Postpone app development * Appeal for mitigating circumstances   Next Meeting:   * Finalise details via emails |