Person in Context Assessment

Project Requirements and Specifications

WSU Psychology Clinic



Shaylin Smith, Christian Manangan, Jiaming Chu

10/13/24

**TABLE OF CONTENTS**

[I. Introduction 3](#_Toc180345784)

[II. Background and Related Work 3](#_Toc180345785)

[III. Project Overview 4](#_Toc180345786)

[IV. Client and Stakeholder Identification and Preferences 5](#_Toc180345787)

[V. System Requirements Specification 6](#_Toc180345788)

[V.1. Functional Requirements 6](#_Toc180345789)

[V.1.1. PICA Assessment 6](#_Toc180345790)

[V.1.2. Phone Application 7](#_Toc180345791)

[V.2. Non-Functional Requirements 8](#_Toc180345792)

[V.3. Use Cases 10](#_Toc180345793)

[V.4. User Stories 14](#_Toc180345794)

[V.5. Traceability Matrix 15](#_Toc180345795)

[VI. System Evolution 15](#_Toc180345796)

[VII. Project Solution Approach 17](#_Toc180345797)

[VIII. Architecture Design 17](#_Toc180345798)

[VIII.1.1. Overview 17](#_Toc180345799)

[VIII.1.2. Subsystem Decomposition 18](#_Toc180345800)

[IX. Data Design 25](#_Toc180345801)

[X. User Interface Design 26](#_Toc180345802)

[XI. Glossary 26](#_Toc180345803)

[XII. References 26](#_Toc180345804)

[XIII. Appendix-1 27](#_Toc180345805)

# Introduction

Mental health has become one of the most pressing issues worldwide, with the World Health Organization listing mental health issues as the leading cause of global disease burden. Whether it is depression, anxiety or other mood disorders, these issues have a profound impact on individuals' daily lives, work productivity and interpersonal relationships. As people's awareness of the importance of mental health continues to increase, more and more technological solutions are being used to help individuals better manage and monitor their emotional health. According to the Global Mental Health Market Study, the demand for digital mental health solutions is expected to grow rapidly in the next few years, driving further development of mental health interventions around the world.

The combination of technology and mental health interventions provides individuals and clinicians with unprecedented opportunities to obtain real-time, personalized feedback and use data-driven approaches to improve treatment outcomes. However, there is still a lack of research on how personality affects treatment response. Although some studies have shown that patients' personality or temperament may affect their response to treatment, more specific evidence is needed to support this claim. Therefore, our team is committed to further verifying this hypothesis through this project - taking patients' personality traits into account can help develop more targeted treatment plans, thereby improving treatment effectiveness.

We aim to enhance the user experience by further developing and integrating two existing applications. The first is a web survey based on Qualtrics that generates personalized psychological assessment feedback for users. The second is a mobile application that allows users to self-monitor emotional events. Our goal is to optimize these apps so that they not only provide clear and timely feedback, but also improve the contextual recommendation capabilities of self-monitoring apps through more sophisticated algorithms. Ultimately, our team aims to integrate these two systems into a more comprehensive mental health support platform that helps users better manage their emotions and behaviors while providing clinicians with deeper insights.

# Background and Related Work

Before we assess current industry researchers in our domain, we need to establish the domain in which our project is going to operate. We define the domain as “interactive personality assessments and self-monitoring tools used for clinical therapy.” The conceptual foundation of this project is based on serval key studies about personality assessment.

The Person in Context Assessment (PICA) provides the project’s main structure and is the first foundational work. PICA is an application of social cognitive theory, specifically Cervone’s model. PICA evaluates personality traits in the context of social interactions, focusing on variables like beliefs, goals, self-schema, and temperament. The Qualtrics survey report and web-based app are structured with PICA as the framework [1].

The Relational Self-Schema Measure (RSSM) is the second piece of relevant work. RSSM looks at how people describe themselves with the four or six most important or frequently interact or think about. PICA does not directly integrate RSSM but utilizes the concepts of RSSM to understand what relational context impacts a person’s self-concept. The project’s depth of personality evaluations is further enhanced by incorporating relational schema concepts. The project provides personalized feedback not just based on the individual but also considering the relation of key figures in the individual’s life [1].

Finally, the BIS/BAS scales, which measure behavioral inhibition and behavioral activation respectively, is the third piece of key influence. Both scales regulate how people respond to potential incentives and punishments. This makes both useful for understanding motivational systems. The principles of BIS/BAS in the feedback that it generates are shown though the web-based application, keeping track of participants’ daily emotional events and assists them in understanding how their personality traits influence their responses to different stimuli [1].

Although PICA serves as the core personality assessment model in our project, the additional insights derived from RSSM and BIS/BAS help us better understand overall personality and behavior. This gives us and clinicians a clearer picture of how different people behave and manage their emotions.

Several technical skills and tools will be required to complete this project. Python is the main language used to power the core functionality of the Qualtrics survey report generator and web-based monitoring app. Furthermore, since the project has been deployed on Python Anywhere, we will need to continue using and managing the web application using that platform. In order to manage the potential vast amounts of personality and behavioral data that our app collects, we will need to be skilled in MongoDB. As for the frontend, we will need proficiency with the relevant knowledge and technology related to web application development. Finally, integrating AI-driven clustering approaches using ChatGPT will require the learning of building applications with generative AI, which will improve the app’s ability to provide users with even more personalized insights and suggestions.

# Project Overview

There are many different types of psychological treatment for those with mental health issues and it can be difficult to figure out what treatment best suits each person. One way to approach finding the best treatment for a patient is by factoring in the person’s personality and looking into how their temperament affects their thoughts, emotions, and behavior in different contexts. The WSU Psychology Clinic is trying to utilize this personality-based approach and investigate a patient’s temperament and responses to everyday situations and try to build the best treatment based upon those results. Through this project, the WSU Psychology Clinic can adopt a more evidence-based approach and show how a person’s personality effects how well they respond to different types of treatment.

This is where the Person in Context Assessment (PICA) comes into play. The focus of this project is split among two separate Qualtrics surveys used to collect more information about a patient’s temperament. The first survey is one that asks questions about if the user has any goals in different areas of their life such as work, relationships, and personal health as well how much the user is working towards that goal and how happy they are about it. Then the results of the user’s temperament are calculated, and a PDF of the results are sent to the email. Our first course of action will be to fix the previous Python code that emails the PDF results to the user. Right now, the user must go into PythonAnywhere to be able to fetch their results, but we want to make sure that their results can be emailed to them. As well, we will be looking into editing the format of the PDF and changing the graphs and results to make sure that it is understandable to the user and looks visually appealing.

The second survey that is used in PICA is one that is used by the patient daily. The user will input their experiences that day and then will also fill in their thoughts, feelings, goals, and actions for the event. Then, emotionally similar events are grouped together and then the clinician can look at both the patient’s temperament as well as how they reacted to these events and use that personality-based data to give a more personalized mental health treatment plan. With this second survey, we want to work on integrating the first survey so that the user would be able to see their original PICA results inside of this daily survey. The clinician should be able to enter the patient’s PICA results into the daily survey and have the user be able to see the PDF of their results in the daily survey app. As well, the daily survey should be able to tailor questions to the user based upon their temperament results and better predict how they may react to daily situations.

Finally, with the daily survey we will be looking into creating a more complex clustering algorithm. We need to be able to group a user’s experiences based upon how similarly the patient thought or felt during the event and that is done based upon what feelings the patient fills in on a checkbox list. Any events that have over 50% shared feelings will then be categorized as the same. We want to either move towards using more complex math to group the different experiences more accurately or possibly investigate utilizing the ChatGPT API to cluster the data. The user can fill in the checkboxes for how they were feeling but they can also write any specific thoughts or emotions into a textbox. The current simpler clustering algorithm does not factor this text box in when clustering so the patient or clinician would have to go in themselves to the survey and read the text box and manually cluster the events if they see the need to. By utilizing AI and ChatGPT, we would remove the need to manually read and cluster those events that have information stored in the text box and instead the AI could read the text boxes and suggest which events should be clustered.

# Client and Stakeholder Identification and Preferences

Our primary customer is the Washington State University (WSU) Psychology Clinic, where the project will be used to help clinicians better understand their patients' personality traits and provide personalized treatment recommendations. Our mentors, Dr. Scott and Belinda Lin, are the primary contacts for the project and will provide guidance on how to optimize the personality assessment feedback and self-monitoring applications. The project will ultimately be used by clinical graduate students and clients at the WSU Psychology Clinic as a tool to assess the relationship between patient personality and treatment response. In addition to WSU, we have established potential partnerships with the Portland Psychology Clinic, where clinicians are also interested in using this system to help their patients.

Potential customers include the Portland Psychology Clinic and other psychotherapy institutions. To attract the interest of these institutions, our software project needs to ensure easy integration and use processes. The interface of the personality assessment report and self-monitoring application must be clear and easy to use, report generation and data processing must be efficient, and the feedback content can directly support clinical decision-making. To further promote application, the project should have cross-platform compatibility so that these institutions can easily deploy the system into their existing technical infrastructure.

Finally, all stakeholders in the project will benefit from a system with a clear structure, concise code, and easy extensibility. This will help clinicians better integrate these tools into their daily work and provide a solid foundation for future system improvements. Our development team will prioritize the needs of Dr. Scott's team, but throughout the design and development process, we will also pay attention to how to meet the needs of other potential customers and institutions to ensure the broad applicability and sustainability of the project.

# System Requirements Specification

This section outlines the key functional and non-functional requirements, use cases, user stories, and traceability matrix for our system. It gives a thorough summary of the system’s objectives, behaviors, and how users interact with it.

## Functional Requirements

Each functional requirement is listed below with a detailed description, source, and priority level.

### PICA Assessment

**Results PDF:**

|  |  |
| --- | --- |
| Description | The PICA Qualtrics survey needs to be able to create a PDF of the resulting charts and graphs that is easy to understand for the user as well as adding more in-depth details and a radar chart. |
| Source | Required from Client. Adding on to already implemented functionality. |
| Priority | Priority Level 0: Essential and required functionality |

**Automatic PDF Sending:**

|  |  |
| --- | --- |
| Description | The PICA Qualtrics survey results will be automatically sent to the email that the user provided in the survey and will be sent after the survey is completed. |
| Source | Required from Client. Client needs to be able to send results to patient. |
| Priority | Priority Level 0: Essential and required functionality |

**Data Collection:**

|  |  |
| --- | --- |
| Description | The PICA survey data from all surveys taken must be sent to the MongoDB database to be stored. |
| Source | Internal requirement from team. Building upon functionality added by previous team. |
| Priority | Priority Level 0: Essential and required functionality |

### Phone Application

**PICA Assessment Results Integration:**

|  |  |
| --- | --- |
| Description | The PICA data for the patient must be able to be inputted into the phone app which allows the user to see their results in the phone app. |
| Source | Requirement from Client. Client needs to be able to integrate the PICA data with the phone app. |
| Priority | Priority Level 0: Essential and required functionality. |

**PICA Based Questions:**

|  |  |
| --- | --- |
| Description | The phone app will ask specific questions based upon the patient’s temperament that was entered into the survey. |
| Source | Requirement from Client. Client needs to be able to ask questions based upon the entered temperament. |
| Priority | Priority Level 0: Essential and required functionality. |

**Clustering Algorithm:**

|  |  |
| --- | --- |
| Description | The algorithm that is used to group events that are “functionally equivalent” based on shared thoughts, feelings, or behaviors will be grouped together either through a mathematically complex algorithm or AI and will be then approved to be clustered by the user or the doctor. |
| Source | Internal requirement from team. Building upon already implemented functionality. |
| Priority | Priority Level 2: Extra features or stretch goals |

## Non-Functional Requirements

Non-functional requirements outline the operational qualities of the system, such as performance, system availability, maintainability, and security, and require accuracy and security of data stored in the MongoDB database. In addition, the system needs to ensure user experience to ensure that it meets quality standards beyond core functionality. Details of non-functional requirements are as follows.

|  |  |  |
| --- | --- | --- |
| **Non-Functional Requirements** |  | **Description** |
| [NFR-1]System Performance |  | The system should be able to scale as the number of users grows, especially when a large number of clinical graduate students or psychological clinics use the system at the same time. |
| [NFR-2]Data Storage and Management |  | All emotional events, evaluation data, and user feedback should be accurately stored in a MongoDB database, and the integrity of the data should be ensured. The database should support automatic backup and recovery mechanisms to ensure that no data is lost in the event of a system failure. |
| [NFR-3]Security |  | The system should have role-based permission management capabilities to ensure that only authorized users (such as clinicians and researchers) can access specific assessment results and user emotional events. |
| [NFR-4]System Availability |  | If the system fails, it should have automatic recovery and error handling mechanisms to ensure that the user experience is not affected. For example, when PDF generation fails, the system should automatically retry or send a notification to the administrator. |
| [NFR-5]User Experience |  | The application should have an intuitive user interface so that users can start using the system without having to read a lot of documentation. The emotional event recording and feedback functions in the mobile self-monitoring application should be simple and easy to understand, and participants can complete them quickly. |
| [NFR-6]Maintainability |  | The system should be designed with an extensible and maintainable architecture to support the addition or modification of future functions. The development team should be able to quickly locate and fix potential problems in the system and update it without affecting users. |

## Use Cases

The use cases describe common scenarios of user interactions with the system, explaining how various functional requirements are applied in specific situations.

A diagram of a person's work flow

Description automatically generated

*Figure l: Use case diagram*

The description of each use case is given below.

**Use Case 1: Complete Survey**

|  |  |
| --- | --- |
| Use Case | Complete Survey |
| Actors | Participant |
| Pre-condition | Participant logged in and started the Qualtrics survey |
| Post-condition | The participant completed all survey questions and the system received the complete survey results. |
| Main Flow | - Participant opens the Qualtrics survey.  - Participant answers all questions and submits the survey.  - The system confirms the survey is complete and ready to generate feedback. |
| Alternative Flow | -If a participant drops out of the survey midway, the system saves the current progress and allows the participant to continue later. |
| Related Requirements | FR1:Results PDF  FR2:Automatic PDF Sending  FR3:Data Collection |

**Use Case 2: Generate Personalized Feedback**

|  |  |
| --- | --- |
| Use Case | Generate Personalized Feedback |
| Actors | System |
| Pre-condition | The survey was completed and the system received the participant's survey results. |
| Post-condition | Personalized assessment feedback was generated, including preliminary feedback and detailed feedback. |
| Main Flow | -The system receives and analyzes survey data from participants.  -The system generates preliminary feedback, including personality assessment results.  -The system generates detailed feedback, providing specific treatment strategies. |
| Alternative Flow | -If the data is incomplete, the system will generate feedback containing only basic information. |
| Related Requirements | FR1:Results PDF  FR5:PICA Based Questions  FR6:Clustering Algorithm  FR7:Data Collection |

**Use Case 3: Generate and Send PDF Report**

|  |  |
| --- | --- |
| Use Case | Generate and Send PDF Report |
| Actors | System |
| Pre-condition | Personalized feedback generated. |
| Post-condition | PDF reports were generated and emailed to participants and clinicians. |
| Main Flow | -The system generates a personalized PDF report based on the feedback, including charts and treatment recommendations.  -The system automatically sends the PDF report to the email address provided by the participant.  -The system confirms that the report was successfully sent. |
| Alternative Flow | -If the email fails to be sent, the system will record the failure information and try again. |
| Related Requirements | FR1:Results PDF  FR4:PICA Assessment Results  FR7:Data Collection |

## User Stories

The following user stories outline the specific actions a user can perform within the system, explaining what the user wants to accomplish and the reasoning behind it. Each user story clearly defines the system’s behavior.

**User Story US1: Generate and Send Personalized PDF Report**

As a User, I want my PICA results PDF to be automatically sent to my email so I can have a personal copy of my results.

Feature: Automatic PDF generation and sending

Scenario: User gets report

Given that the user has finished the PICA fully

When they click the submit survey button

Then the PDF of the results should be automatically sent to the email they inputted into the survey

**User Story US2: Log Emotional Event**

As a User, I want to be able to log an emotional event in the phone app so I can keep track of emotionally similar events.

Feature: Log Emotional Events

Scenario: User logs an event

Given the user is in the phone app

When they fill out the form to log a new emotional event

Then the daily results will be logged in the MongoDB database with the timestamp of the submission

**User Story US3: View Functionally Equivalent Situations**

As a User, I want to be able to see functionally equivalent situations so I can know what events have been grouped together and were emotionally similar.

Feature: View Functionally Equivalent Situations

Scenario: User views their situations

Given the user is in the phone app

When they select to view their emotional situations

Then a list of their situations is shown and are grouped based upon the clustering algorithm.

## Traceability Matrix

The table below aligns functional requirements with their corresponding use cases and user stories. This ensures that all requirements are accounted for and directly connected to specific user scenarios.

|  |  |  |  |
| --- | --- | --- | --- |
| Functional Requirements | Use Case | User Story | Priority |
| FR-1: PICA Assessment Results PDF | UC-3: Generate and Send PDF Report | US1: As a user, I want my PICA results PDF to be automatically sent to my email | Level 0 |
| FR-2: Automatic PDF Sending | UC-3: Generate and Send PDF Report | US1: As a user, I want my PICA results PDF to be automatically sent to my email | Level 0 |
| FR-3: Data Collection | UC-7: Collect User Feedback | US6: As an admin, I want to be able to keep user’s data in a database | Level 0 |
| FR-4: PICA Assessments Results in Phone App | UC-6: Import Personality Assessment Results | US5: As an admin, I want to be able to import a User’s PICA results into the phone app | Level 0 |
| FR-5: PICA-Based Questions | UC-4: Log Emotional Event | US4: As an admin, I want the PDF results of the PICA to be optimized | Level 0 |
| FR-6: Clustering Algorithm | UC-5: View Functionally Equivalent Situations | As a user, I want to see functionally equivalent situations | Level 2 |
| FR-7: Data Collection | UC-4: Log Emotional Event | US6: As an admin, I want to be able to keep user’s data in a database | Level 0 |

# System Evolution

As our project progresses, it is important to keep in mind the fundamental assumptions that drives our development. We must also consider the potential changes that may come up due to evolving technology, changing user expectations, and unforeseen challenges. We will be able to make wise design choices that ensures the system’s longevity, scalability, and adaptability by understanding and anticipating these changes.

Our system’s development relies on several key assumptions about the hardware, software, and user requirements. To start, our current technology stack, which consists of primary Python as well as MongoDB, and Qualtrics, will continue to be maintained and developed in a manner that keeps it compatible with our project. Python Anywhere is a reliable platform for hosting applications and we expect this reliability to continue. Additionally, we anticipate that the end user’s devices will be able to meet the web-based application’s requirements in terms of processing speed and picture quality. The system is designed prioritizing ease of use, as we believe users will need intuitive and easy access to their personality assessment data and daily emotional tracking.

We anticipate several technological and user-driven changes over time that may have an impact on the system. One of which is that as technology evolves, so will cyber threats. Since our system handles sensitive personality assessment data that will be used for research, it must adhere and abide by strict data protection standards. As for the user side, we anticipate that user demands will change as more people use digital tools to manage their mental health. The use of artificial intelligence (AI) may make users demand more personalized and immediate feedback.

We must also consider several risk points that may come up throughout the development of our project. The integration of Qualtrics and the web-based app may not scale well as the number of users increases dramatically. The system now, uses relatively simple algorithms that process and display personalized data. As more users interact with the app, performance issues with data handling and processing could arise.

The app’s current clustering feature that uses a simple rule-based algorithm to determine “functionally equivalent” emotional events is another potential risk. As we explore the use of ChatGPT and other models that could be pre-trained and fine-tuned to fit this project, there is a chance that these approaches will not work as planned. They could also introduce complications and complexity that are difficult to handle in the current system. If there are changes in availability or cost, it will require us to adapt quickly or look for other options.

To address these risks, we are implementing several precautionary measures to mitigate these risks. First, we need to enhance and manage the system with scalability in mind. This is to ensure the system can handle increased user traffic without significant performance issues. As for integrating AI, we will monitor AI technologies and remain flexible in our approach. If ChatGPT or some other model becomes too costly or unavailable, we will investigate other alternative clustering methods. This may include refining the current clustering algorithm or implementing open-source AI tools. Finally, we will create modular system components that will allow us to upgrade or swap out certain features. Doing so will not require a complete redesign of the system.

Our proactive approach to anticipating and adjusting to these changes will be vital as our project progresses. By keeping a close eye of the assumptions underlying our system’s design, and continuously evaluating potential risks and opportunities for improvement, we ensure that we are well prepared to deliver a robust and secure application. This approach allows us to remain flexible and responsive which sets up the project for long-term success and stability.

# Project Solution Approach

The purpose of this section of the document is to outline the architecture, a bird’s eye view of the system, as well as components and system functionality for the project. This section is intended for the WSU Psychology Clinic, project developers, and stakeholders involved. It will serve as a guide for the continuing development and integration of the existing survey and self-monitoring application.

# Architecture Design

# Overview

The current project builds upon the architecture and development efforts of previous iterations, focusing on integrating and improving two primary applications: 1) a Qualtrics-based web survey report that generates personalized personality assessments, and 2) a mobile self-monitoring app that logs emotional events daily. Our architectural design follows the client-server model, which ensures scalability, modularity, and reliable interaction between the client-side user interfaces and the server-side logic that handles data processing, storage, and analysis.

The client-server model was chosen for several reasons. First, this pattern allows for clear separation of concerns: the client focuses on user interaction and inputs, while the server handles data processing, sentiment analysis, and storage. Given the large volume of data and the need for event-based functionality (logging, monitoring, and real-time feedback), this model ensures efficient data handling and user experience. It also facilitates the integration of advanced features, such as treatment recommendations and clustering algorithms in the mobile app, making it flexible for future enhancements.

The system consists of several key components that work together to provide real-time, personalized feedback to users. On the client side, users interact with the Qualtrics-based survey and self-monitoring mobile app to input their emotional events or complete personality assessments. These inputs are processed by the server, which stores the data securely, analyzes it using sentiment analysis and other algorithms, and generates relevant feedback. The feedback is then delivered back to the client as personalized reports or recommendations.

The UML components illustrate the interaction between client-side (UI) and server-side components (application logic, DBMS, and security), as well as the flow of data in the system. The diagram highlights how different modules interact to achieve the goals of the system, from user input to data processing and feedback generation.

**A diagram of a system

Description automatically generated**

*Figure II: Architecture Design*

# Subsystem Decomposition

**2.1. [UI Handler]**

**2.1.1. Description**

The UI Handler is responsible for managing all interactions between the user and the system. It handles input from both the web-based Qualtrics survey and the mobile self-monitoring app. This subsystem ensures that users can submit their survey responses and emotional event logs, and it relays this information to the appropriate back-end components for further processing. It also manages the output by displaying personalized feedback or reports generated by the system.

**2.1.2. Concepts and Algorithms Generated**

The UI Handler does not utilize complex algorithms but instead focuses on input validation and user interaction. It uses basic validation mechanisms to ensure that the input data (survey responses, logs) is in the correct format and passes the necessary checks before being sent to the system. The selection of this solution was driven by the need for an efficient and responsive user interface that can process user inputs in real-time and provide an intuitive experience.

**2.1.3. Interface Description**

**Services Provided:**

|  |  |  |
| --- | --- | --- |
| Service Name | Service Provided To | Description |
| Input Collection | Input Handler | Collects and validates user input (survey data or emotional events) and forwards it to Input Handler. |
| Display Feedback | Application Logic | Displays feedback or reports generated by the system back to the user. |

**Services Required:**

|  |  |
| --- | --- |
| Service Name | Service Provided From |
| Processed Feedback | Application Logic |
| Input Validation | Input Handler |

**2.2. [Environmental Handler]**

**2.2.1. Description**

The Environmental Handler manages external data interactions, ensuring that the system can adapt to the environment it operates in. It gathers data from sources such as mobile sensors, location data, and other external APIs, which is used to enhance the user experience in the self-monitoring app. For example, it might adjust recommendations based on the user's environment.

**2.2.2. Concepts and Algorithms Generated**

The Environmental Handler utilizes location-based algorithms to detect user context and adjust app behavior. It uses a combination of external APIs and real-time data gathering techniques. The decision to use these algorithms was based on the need to provide a dynamic and context-aware experience for users, especially in the self-monitoring app.

**2.2.3. Interface Description**

**Services Provided:**

|  |  |  |
| --- | --- | --- |
| Service Name | Service Provided To | Description |
| Location Data | Application Logic | Provides location-based information to adjust app behavior or user feedback. |
| External Data Updates | Input Handler | Supplies data from external APIs, like weather or location data, to augment user inputs. |

**Services Required:**

|  |  |
| --- | --- |
| Service Name | Service Provided From |
| Input Processing | Input Handler |
| External API Data | External Sources |

**2.3. [Input Handler]**

**2.3.1. Description**

The Input Handler processes all user inputs submitted through the UI. It ensures the correct format and validity of the data before sending it to the Application Logic for further processing, such as report generation or emotional analysis.

**2.3.2. Concepts and Algorithms Generated**

The Input Handler employs data validation algorithms to ensure that all inputs conform to the expected format and values. This was chosen to maintain the integrity of data entering the system and prevent errors during further processing.

**2.3.3. Interface Description**

**Services Provided**:

|  |  |  |
| --- | --- | --- |
| Service Name | Service Provided To | Description |
| Validated Input | Application Logic | Sends validated user inputs (survey or emotional logs) for further processing. |
| Data Preprocessing | Sentiment Analysis Algorithm | Prepares user input data for sentiment analysis. |

**Services Required**:

|  |  |
| --- | --- |
| Service Name | Service Provided From |
| User Input Collection | UI Handler |
| Environmental Data | Environmental Handler |

**2.4. [IDE Reader]**

**2.4.1. Description**

The IDE Reader collects data from the Integrated Development Environment (IDE) or external file sources, and supplies it to the relevant system components, such as for analysis or report generation.

**2.4.2. Concepts and Algorithms Generated**

No specialized algorithms are used here beyond standard file reading and parsing techniques. The primary objective is to accurately retrieve and transfer data.

**2.4.3. Interface Description**

**Services Provided**:

|  |  |  |
| --- | --- | --- |
| Service Name | Service Provided To | Description |
| Data Input | Application Logic | Reads data from files or external sources and forwards it for processing. |

**Services Required**:

|  |  |
| --- | --- |
| Service Name | Service Provided From |
| Data Storage Access | Database Management |

**2.5. [Database Management System]**

**2.5.1. Description**

The Database Management System (DBMS) stores all user data, including survey responses and logs from the self-monitoring app. It ensures that data is efficiently stored, retrieved, and updated as needed, while providing security measures to protect user privacy.

**2.5.2. Concepts and Algorithms Generated**

The DBMS uses standard database indexing and encryption techniques to handle large volumes of data securely and efficiently. The trade-off considered was between performance and security, resulting in the selection of algorithms that prioritize data integrity and protection.

**2.5.3. Interface Description**

**Services Provided**:

|  |  |  |
| --- | --- | --- |
| Service Name | Service Provided To | Description |
| Data Storage | Application Logic | Provides secure storage for user inputs and logs, as well as survey results. |
| Data Retrieval | Sentiment Analysis Algorithm | Retrieves relevant user data for analysis. |

**Services Required**:

|  |  |
| --- | --- |
| Service Name | Service Provided From |
| Processed Data Storage | Application Logic |
| Data Access | IDE Reader |

**2.6. [Sentiment Analysis Algorithm]**

**2.6.1. Description**

This subsystem performs sentiment analysis on the data collected from users. It evaluates survey responses and daily logs, extracting emotional patterns and providing insights that are used in generating feedback and recommendations.

**2.6.2. Concepts and Algorithms Generated**

The sentiment analysis algorithm relies on natural language processing (NLP) techniques to extract emotional cues from user inputs. This allows for personalized feedback based on users' emotional states. The algorithm chosen was based on its accuracy and ability to handle large-scale input data efficiently.

**2.6.3. Interface Description**

**Services Provided**:

|  |  |  |
| --- | --- | --- |
| Service Name | Service Provided To | Description |
| Sentiment Analysis | Application Logic | Provides emotional analysis based on user inputs, assisting in generating personalized reports. |

**Services Required**:

|  |  |
| --- | --- |
| Service Name | Service Provided From |
| Input Data | Input Handler |

**2.7. [Security Component]**

**2.7.1. Description**

The Security Component handles the security of user data throughout the system, providing encryption, authentication, and secure communication between subsystems to ensure that all user interactions and data storage remain protected.

**2.7.2. Concepts and Algorithms Generated**

The Security Component uses encryption algorithms (such as AES) and authentication protocols (OAuth) to ensure secure access and data transmission. Trade-offs between performance and security were considered, with a focus on safeguarding sensitive user data without significantly impacting system performance.

**2.7.3. Interface Description**

**Services Provided**:

|  |  |  |
| --- | --- | --- |
| Service Name | Service Provided To | Description |
| Data Encryption | Database Management | Ensures all stored data is encrypted and secure from unauthorized access. |
| User Authentication | UI Handler | Manages login credentials and ensures that only authorized users can access the system. |

**Services Required**:

|  |  |
| --- | --- |
| Service Name | Service Provided From |
| User Data Encryption | Database Management |

# Data Design

Currently we are storing our databases in MongoDB Atlas that was established by the previous team. This allows the client to be able to have access to the database and easily add our team and any future teams to the project. This MongoDB database holds all our data related to the web app that collects the user’s daily logs of their thoughts and feelings. In this database we store users, surveys, signatures, situations, and all multiple-choice options as collections in the database. Each survey document contains an ID, user, signature, fi-then, situation, positive feelings, negative feelings, positive thoughts, negative thoughts, and behavior. The web app then makes a connection to the cluster of our database and then is ready to use. The positive and negative thoughts and feelings as well as behavior options each are stored in the database and then appear as multiple-choice options in the app. The web app then retrieves these multiple-choice options from the database and displays them on the screen when a person is logging into an event. The document classes for these multiple-choice options are stored in Models.py with each collection in Mongo having a class in the main app file psychclinic.py which fills the collections when the app is initialized. Once the app is run, the multiple-choice options are filled out and do not need to be filled out again. If we need to update the multiple-choice options, we can easily do so through MongoDB by adding them into the database.

Another way we store our data is through a csv file we create when retrieving the survey data from Qualtrics, which hosts the PICA. When we call Report\_Generator.py after someone completes their PICA, the report generator then connects to the Qualtrics API to collect the list of surveys taken which is then locally saved as a csv file in PythonAnywhere. That csv file is then loaded into our report generator as a DataFrame. We take the last row of the DataFrame, which is the data of the most recently taken survey and use that data to generate our report and send it to the corresponding email. Storing the data this way allows us to have local access to the list of surveys and potentially send multiple survey reports to multiple people based off the one csv file.

# User Interface Design

The user interface was originally designed and developed by a capstone team prior to the previous one and we have adhered to the same color schemes and icons they created. The previous capstone team made updates to the home page which includes icons for each page in place of text. They also changed the dashboard to only display home and logout. Images of the user interface can be found in the appendix.

Looking at the images in the appendix, the login screen (Image 1) is still the same as it was when the capstone team prior to the previous one created it. Users have two options when creating an account: an admin account or a regular user account. After entering the required information and creating credentials, they can log in. The previous team discussed this login page with the client, and it was concluded that no changes were necessary. The user’s name is displayed after logging in which is based on information from their login (Image 2). The team prior to the previous updated the landing page which allows users to start a new journal entry. The user can then begin a journal entry by selecting either a positive or negative entry. This takes them through additional pages with relevant questions and prompts (Image 3). Reviewing past journal entries is another important function (Image 4). From the admin’s perspective, admins have access to see every user’s survey. Admins have the option to view surveys and utilize the Qsort tool after signing in (Image 5). The “view survey” option lets them view surveys for all users (Image 6). The other option is Qsort which allows admins to assign or insert a survey to specific users.

Many use cases will utilize these interfaces for user interaction. Use case 4: Log Emotional Event, the journal entry pages are vital for users to log their emotional events. Use case 5: View Functionally Equivalent Situations, the past journal entries page will be used when users want to view similar emotional situations that were determined by the system. Use case 6: Import Personality Assessment Results, the admin login as well as the survey views will be used by clinicians to important and manage personality assessment results.

# Glossary

**ChatGPT**: Stands for Chat Generative Pre-Trained Transformer. It is a large language model-based chatbot that allows users to refine and steer a conversation towards desired length, format, style, etc.

**Qualtrics**: Qualtrics is a cloud-based platform that allows users to create, distribute, and analyze surveys. It is the platform used to host the two surveys of the project.

# References

|  |  |
| --- | --- |
| [1] | W. D. Scott, S. Paup and C. Kirchhoff, "Clinical Application of Social Cognitive Theory: A Novel Personality Assessment Procedure and a Case Study of Personality Coherence," *European Journal of Personality,* vol. 36, pp. 371-390, 2022. |

# Appendix-1

**Additional User Stories**

**User Story US4: Optimize PDF Report Content**

As an Admin, I want the PDF results of the PICA to be optimized so that the Users are easily able to understand their results

Feature: Optimized PDF Report Content

Scenario: User reads their PDF results

Given the user has taken the PICA

When they receive their report

Then they will be able to clearly understand the results and understand what their temperament means.

**User Story US5: Import Qualtrics Assessment Results**

As an Admin, I want to be able to import a User’s PICA results into the phone app so that the User’s temperament can be factored into the phone app’s questions.

Feature: PICA Phone App Integration

Scenario: Admin enters PICA results

Given I am an Admin, and a user has taken the PICA

When I enter the User’s results into the phone app

Then the questions in the daily survey will be more tailored to the person based upon their temperament.

**User Story US6: Collect User Experience Data**

As an Admin, I want to be able to keep user’s data in a database so that I have a record of the user’s PICA and phone app responses.

Feature: User Data Collection

Scenario: User completes PICA

Given I am an Admin and at least one person has completed the PICA

When I access the MongoDB database

Then I am able to see the user’s results to the assessment and can query the database.

**Additional Use Cases**

**Use Case 4: Log Emotional Event**

|  |  |
| --- | --- |
| Use Case | Log Emotional Event |
| Actors | Participant |
| Pre-condition | Participants were logged into a mobile phone self-monitoring application. |
| Post-condition | Emotional events and related information have been saved to the database |
| Main Flow | -The participant opens the mobile app and chooses to record an emotional event.  -The participant enters the type of emotion, thoughts, feelings, and behavior information.  -The system saves the record and confirms that it was saved successfully. |
| Alternative Flow | -If the device is offline, the system will automatically sync the data when the connection is restored. |
| Related Requirements | FR5:PICA Based Questions  FR6:Clustering Algorithm FR7:Data Collection |

**Use Case 5: View Functionally Equivalent Situations**

|  |  |
| --- | --- |
| Use Case | View Functionally Equivalent Situations |
| Actors | Participant, System |
| Pre-condition | View Functionally Equivalent Situations |
| Post-condition | The system provides participants with functionally equivalent situational suggestions based on the recorded data. |
| Main Flow | -The system analyzes the emotional events recorded by the participants.  -The system generates functionally equivalent situational suggestions based on a clustering algorithm.  -The system displays the suggestions to the participants. |
| Alternative Flow | -If the system fails to identify similar situations, participants will be prompted to record more events to obtain more precise suggestions. |
| Related Requirements | FR5:PICA Based Questions  FR6:Clustering Algorithm FR7:Data Collection |

**Use Case 6: Import Personality Assessment Results**

|  |  |
| --- | --- |
| Use Case | Import Personality Assessment Results |
| Actors | Clinician |
| Pre-condition | A PDF report of the personality assessment results has been generated and sent. |
| Post-condition | The clinician has entered the assessment results into the mobile app. |
| Main Flow | -Clinician logs into mobile app.  -Clinician manually enters or uploads personality assessment results.  -System confirms data was saved successfully. |
| Alternative Flow | -If the data is incorrect, the system will prompt the doctor to re-enter or modify the data. |
| Related Requirements | FR4:PICA Assessment Results FR6:Clustering Algorithm FR7:Data Collection |

**Use Case 7: Collect User Feedback**

|  |  |
| --- | --- |
| Use Case | Collect User Feedback |
| Actors | Researcher |
| Pre-condition | Participants and clinicians completed the corresponding functional operations. |
| Post-condition | User experience data is stored in the database for subsequent analysis. |
| Main Flow | -Researchers regularly push experience surveys to users.  -Users complete the experience surveys and submit data.  -Researchers review the data for subsequent analysis and improvement. |
| Alternative Flow | -If the user skips the survey, they will be prompted to fill it out again the next time they use the system. |
| Related Requirements | FR6:Clustering Algorithm FR7:Data Collection |

**UI Images**

**A screenshot of a login form

Description automatically generated**

*Image 1: Login*

**A screenshot of a computer

Description automatically generated**

*Image 2: Home Page*

A screenshot of a computer

Description automatically generated

*Image 3: Logging an Event*

A screenshot of a computer

Description automatically generated

*Image 4: Past Situations*

A close-up of a blue and white screen

Description automatically generated

*Image 5: User Surveys*

A screenshot of a computer

Description automatically generated

*Image 6: User Qsort Entry*