Solution & Technology Chapter Export

Overview

Our solution consists of a mental health care app with a chatbot functionality at its core, where users can interactively talk about their problems, get advice, and improve their mental well-being in a safe environment.

In this chapter, we’re covering the solution only from a user and business perspective. The technology and architecture behind the solution will be described in [Chapter X].

Key Features

Intro

The app consists of three key features: chatbot, mood tracking and journaling, where the chatbot builds the core functionality of the app and directly connects with the other two features.

In general, the chatbot provides the main interface for the user and can be used standalone, whereas the mood tracking and journaling are more supportive and optional features for self-reflection and data gathering.

However, especially the data gathering part makes these supportive features very powerful, since they can provide systematically conducted, well-structured data for the chatbot to better personalize the experience, react to the current mood of users, memorize and react to certain events. Additionally, the data can be used by the model to analyze and draw inferences from the whole user base about which kind of problems might be correlated with which kind of behaviors and what questions and suggestions from the chatbot work best to improve the situation of the users.

This means that the chatbot will learn and improve over time the more users there are and the longer they’re using the app.

Chatbot

As mentioned above, the chatbot provides the core functionality of the app, that can be used for a variety of mental-health related use cases depending on the specific needs of the user. This means, that the way how, when and why users interact with the chatbot can be quite different.

Some users might use the chatbot just occasionally to talk about specific problems they have at work or seek for advice in certain situations like anxiety. Others might use it frequently up to a daily basis to constantly reflect and improve their mental well-being or use it as motivational tool to build new habits, etc.

The overall chat experience will be similar to general purpose chatbots like ChatGPT, but specialized on mental health and personalized based on the users data.

In the following, we'll list the three key success factors that are essential for the chatbot feature, briefly describe how we're going to achieve them and which role large language models play in that regard.

1. The chat experience must feel natural and engaging, reaching a level that feels like chatting with a human.

   Our chatbot architecture will be build on top of GPT, a large language model and breakthrough-technology, empowering extremely successful chat applications such as ChatGPT.

2. The chatbot must be an expert in effective mental health coaching and therapy practices and also have the domain knowledge in relevant disciplines like psychology and specific sub-fields, such as stress management, habit building, etc.

   The large language model will be trained and fine-tuned on data sets related to the relevant domains, including scientific publications as well as real conversations between human therapists and patients

3. It must be driven by data from its users to memorize conversations and build personalized experiences.

   The data will be gathered through the conversations itself, the mood tracking and journaling features of the app as well as third-party integrations like health-, fitness- and sleep-trackers

The Importance of Large Language Models

At this point we want to highlight the importance and impact of the technology which is going to leverage our application, whereas the technical part will be described in the technology chapter.

Over the past few months, there's been growing a huge interest in large language models such as GPT-4 from OpenAI, BERT from Google, and LLaMA from Meta, especially through OpenAI's chat application ChatGPT that achieved the milestone of 100 million monthly active users just two months after its launch, making it the fastest-growing consumer application in history. In comparison, TikTok took about nine months to reach the same user base, while Instagram took two and a half years.

Not only the users are fascinated by all the possibilities, but also many experts across a variety of domains believe that the adoption of this technology is likely to mark a breakthrough similar to the introduction of the internet and the smartphone.

Although the technology is still comparably new, there are many potential use cases and applications emerging.

Microsoft, as one of the biggest investors in OpenAI, which is the company behind ChatGPT has already started to implement the technology in form of a “Copilot” among its office applications like Outlook, Word, Excel and PowerPoint as well as Microsoft Edge and Bing, which are empowering billions of people around the world every day. Especially the latter part, the integration into the search and browser is causing a major threat for Google, but also other players, competing in the so called “AI race”. Almost any company, especially in the tech and software sector are currently exploring the potentials and trying to integrate large language models in their product and services. Therefore, as already described in previous chapters, we strongly believe that this technology is going to transform the digital mental health industry as well.

Mood Tracking

The mood tracking feature allows users to regularly track their emotional state, activities and physical symptoms in a very efficient and systematic way.

The process is very simple and - depending on the categories the user want to track - can be done within a minute. Typically the user is using this features on a daily basis to rate their overall mood on a scale from 1-5, select their current emotion from a pre-defined categories (e.g. happy, sad, anxious, etc.) and log specific activities from their day like exercise, work or socializing. Furthermore they can log physical symptoms like headaches and sleeping patterns such as sleep duration and quality.

As mentioned before, on a long-term we’re planning to add third-party integrations to the app, so specific activities like sports, sleeping patterns, etc. could be gathered from health- and sleep trackers automatically and in more detail.

Besides the data gathering part, the mood tracker also includes a dashboard with visualizations for the user to gather data-driven insights for self-awareness and reflection.

However, since most users (especially those who have never used mood trackers before) are not that data-savvy and often not able to draw valuable inferences from the data, the main benefit of the feature of course will be the integration with the chatbot, which will be described in more detail in the chapter below.

Journaling

The journaling feature allows users to write down their thoughts, feelings and experiences in free-form. Even though there are some similarities and overlapping parts between the journaling and mood tracking feature, the journaling follows a slightly different purpose, focusing more on the processing of emotions and finding solutions, rather than tracking and monitoring emotional states and activities in pre-defined categories.

Therefore, journaling can be a very powerful tool for mental well-being, as it triggers processes that are similar to certain therapy practices, providing various benefits, including:

1. Self-awareness: Writing down thoughts and feelings can lead to a greater understanding of one's emotions, patterns, and triggers, fostering self-awareness and personal growth.
2. Emotional processing: Journaling can help users process complex emotions or experiences, serving as a healthy outlet for self-expression.
3. Stress reduction: Writing about stressful or emotional events can help to release tension and reduce stress levels by organizing and making sense of one's thoughts and feelings.
4. Problem-solving and decision-making: Journaling can help users clarify their thoughts, explore different perspectives, and evaluate potential solutions, which can contribute to better decision-making and problem-solving skills.
5. Goal-setting and tracking: Writing down personal goals and monitoring progress can motivate users to stay committed and focused on their objectives.
6. Memory and cognitive function: Regular journaling can help improve memory and cognitive function by encouraging users to recall and articulate their experiences and thoughts.
7. Creativity: Journaling can stimulate creativity by providing an open space for exploring ideas, expressing emotions, and engaging in free-flowing writing.
8. Mindfulness and self-reflection: Journaling encourages mindfulness and self-reflection, as users become more aware of their thoughts, emotions, and reactions in the present moment.
9. Coping skills: Journaling can help users develop healthy coping skills by providing a space to express emotions, identify patterns, and explore alternative ways to manage stress or challenging situations.

10. Communication skills: Regularly articulating thoughts and emotions in writing can improve written and verbal communication skills.

On the other hand, journaling is way more time consuming and often not that easy for the average user to write down their thoughts on a blank space.

Therefore, journaling apps often provide a set of prompts or questions that give a good starting point and guide the user in the right direction, e.g. to write down positive things about that day, which will automatically lead to a more positive mindset over time.

The problem with most existing journaling apps is that most of them just provide a static set of prompts / questions and are not capable (or very limited) in interpreting the users content. That’s where our integrated chatbot solution can provide a big value, because it can not only interpret the journaling input, but also combine it with the conversations and mood tracking data to provide a personalized set of prompts that will adapt to the current situation, needs and goals of the user.

Connection between Chat, Journaling and Mood Tracking

As already described in the previous chapter about the three key features, all of them are designed to integrate with each other and ultimately to provide data for the chatbot to get a more comprehensive understanding to offer personalized recommendations.

For example, through the integration of these features, the chatbot could:

1. Identify triggers and recommend coping strategies: By analyzing the data, the chatbot could recognize situations or activities that consistently lead to negative emotions, provide suggestions to manage or avoid these triggers, and recommend helpful coping strategies, such as meditation, exercise, or socializing, to improve their emotional well-being.
2. Encourage positive habits and personalized goal setting: If the chatbot detects that certain activities or habits contribute to improved mood, it could encourage the user to maintain or increase their engagement in those activities, while also using insights from journal entries and mood tracking data to help users set personalized and realistic goals related to their mental well-being.
3. Offer context-aware support and encouragement: The chatbot could provide motivational messages, affirmations, or empathetic responses based on the

user's mood, journal entries, and specific situations they have journaled about, leading to more meaningful and relevant interactions.

4. Provide insights and facilitate self-discovery: By analyzing trends and patterns in the user's data, the chatbot could offer insights into their emotional well-being, help them make informed decisions, and enable users to gain new understanding of their own emotional experiences, leading to increased self-awareness and personal growth.
5. Early intervention and reinforcement of positive behaviors: The chatbot may detect early signs of emotional distress or potential mental health issues by analyzing mood tracking and journaling data, allowing it to provide timely support, suggest seeking professional help, and identify and reinforce positive behaviors, habits, or coping strategies that are contributing to the user's mental well-being.

We strongly believe that the combination of these features are essential for providing the most effective mental-health solution. Apart from making the chatbot more powerful through these two additional data sources, we also think that offering different ways to interact with our app, will increase the likelihood of consistent and long-term usage. A standalone chatbot might be great for occasional usage, but the mood tracking and journaling routines is likely to increase user retention.

User Journey

As already mentioned, the way how, when and why users interact with our app can be quite different, depending on the specific needs and goal of our users. Therefore, the user journey is varying between different user types.

In general we can differentiate between two main user types:

- Occasional chat user: Although we believe that regular users will benefit the most, depending on the needs of the user, some might just use the chat function occasionally to seek advice or support in certain situations. In that scenario, the user is opening the app, the chatbot will start with a general (still personalized and engaging) prompt to start the conversation. At the end of the conversation, the chatbot will ask the user if he likes to catch up soon and offer him to send a reminder via push notification after a defined period of time. After the conversation, the chatbot will store a summary of the conversation and update structured data about the user in the background so that the chatbot can refer to that in future conversations.

- Power user: These are the users that are using the app on a daily basis to tracking their mood, use the journal and interact with the chat. These users will benefit from the most personalized and effective experience. The typical journey for those users would be to check-in routinely at the same time of the day, e.g. each morning after waking up or at night before going to sleep. The journey starts by tracking the mood and activities. After that, depending on the preference settings of the users, he will get 1 to 3 prompts for journaling. Those prompts will be already personalized based on the mood tracking data and historic data from conversations, mood tracking and journaling. The journal entries will be stored separately from the main chat conversations for easy access for the user later on (e.g. reflection purposes). At the end of the journaling, the chatbot will offer to start a conversation with a personalized message based on the input from the journaling and mood tracking data. In most cases, depending on the day and situation, the user might want to have a short chat with the chatbot. Otherwise the user will just quit the app after finishing the mood tracking and regular journaling part.

Of course, the user types can be segmented in a more fine-granular way, including the respective problems, goals and demographics of the user. In this case we just keep it simple for demonstrating the main user journeys.

Furthermore, apart from the regular app usage, we also plan to integrate an onboarding dialogue for first-time users. Within this dialogue we want to gather data about the demographics, personal goals, experience with mental health apps and his mental health condition.

This data will be used for kick-starting the personalization of the experience, as well as for assessing the mental state of the user for safety reasons. Especially the latter part is very important, because our app is not designed for treating patients with serious mental-health disorders. Therefore, depending on the mental-health assessment, the chatbot will be more or less restrictive in its conversations. In case the chat detects any suspicious behavior in that regard, it will suggest the user to seek professional help from a human therapist.

Long-Term Features

Apart from the chatbot, mood tracking and journaling features we’re also planning to roll out additional features and functionalities in later product iterations, briefly described in the following:

Connection to Third-Party Apps

As already mentioned, we're planning to offer the user to connect third party apps such as sleep-, health- and fitness-trackers to gather further relevant data about the behavior of the user in an automated way.

On top of that we also want to integrate open data, such as weather information to get a better understanding of the environment and external factors of the users, which might influence his mental health state and he might not even be aware of.

Both additional data sources could be extremely helpful for the model to detect patterns, gather new insights and improve the chatbot over time.

Let's make a simplified example for demonstration purposes: A user starts to complain regularly about an increasing stress-level at work that affects his sleep quality and mental well-being. At the same time, through the connected sleep- and fitness-trackers, the model detects that he stopped exercising three weeks before he mentioned these issues for the first time. It will also infer from the sleep-tracker data that one week after quitting workouts, his sleep duration and quality got worse. As a result, the chatbot will show the user these correlations, explain that his workouts in the past probably helped him to relieve stress, better control thoughts about work in his private life and finally suggest him to start exercising again. When the user continues exercising, the stress and sleep-related metrics are improving again.

What's even more promising is that the model will be able to detect such patterns and apply this knowledge in similar cases of thousands of other users, learn which behaviors affect peoples mental well-being and which measures are most effective. Furthermore, it will be able to pro-actively inform the users once it detects such behavioral changes, so it can prevent users from having those issues in the first place.

Voice-based Conversations

In the beginning the conversations will be text-based only. However, due to recent technological advancements in generative AI there have been huge step towards computer generated voice that sounds human-like and even can imitate specific people in a very realistic way.

There are multiple benefits by implementing voice-based conversations:

1. We make the app more accessible, especially for users with disabilities that often suffer more from mental well-being and therefore potentially could benefit the most.

2. Through voice-input from the users the model can derive additional information about the user, how he is feeling, etc. that are otherwise hidden through text input. This will help the model to adapt and personalize the conversation even further.
3. It’s much easier and natural to talk than to write. Writing takes more time and also makes people over-think and rephrase their content, thus leaving out potentially relevant information.
4. Voice makes the computer to be perceived as more empathic, at least when it sounds realistic like recent models. Furthermore, different people feel more trust with different voice tones and styles. Users will be able to choose the one they feel most comfortable with.

Connection with real therapists

At a later stage we also intent to collaborate and integrate human therapists into the platform and implement a matching mechanism for users to reach out to therapists that suit them best.

This will allow users to talk to an expert in specific cases or just for regular check-ups in an easy way without having all the administrative efforts, etc. This could be interesting especially for the users with more serious mental-health issues, as for those the chatbot experience will be more limited due to the safety reasons mentioned earlier.

For therapists this collaboration could be very interesting as well, since they’ll be able to reach a new audience - people who didn’t want to go to a therapist before, but now after using the app having a better understanding about mental-health care and experience benefits of it.

Additionally we want to provide a set of features to the therapists including a chatbot to get better insights and make them more effective. For example, the users will be able to share selected data to therapist including a chatbot-generated summary about the situation of the user. Also, we want to allow them to record their sessions and create a transcript with voice-to-text technology which will be accessible through the chatbot. This will allow therapists to get a quick summary before the next session, ask about specific things from previous sessions and even get some data-driven recommendations from the chatbot before, after or even within therapy sessions.

For our business there are also multiple benefits from this collaboration. First, we will be able reach a broader audience, including people with more serious mental-health

issues that are or potentially would like to go to therapy. Second, we're creating a new revenue stream, since we plan to take a commission fee for each session between user and therapist. And third, which will be the biggest benefit in long-term, through the access to the recorded therapy sessions, we'll be able to train and improve our model.

Pricing

Our main revenue stream will be based on a subscription model for the chat functionality. We'll going to offer a 14 days free trial for every new user and offer a basic version of the supportive features (mood tracking & journaling) as well as one limited chat session per month for free. This free usage option is required on the one hand to align with our purpose to democratize personal health support by lowering the barriers for everyone. On the other hand, this will also allow us to increase user retention, convert some of the users into paid users over time and last but not least to gather more data that will help us to get crucial insights and improve our algorithms over time.

The paid subscription will be X EUR per month. The paid users will get full access to all features and a credit of X chat messages per day, which will be more than enough for most users. Unused credits will be taken over to the next day. The limitation is needed to maintain a good performance of our services and to prevent us from huge costs generated from some extreme users or bad actors.

Technology

Introduction

In this chapter we'll describe the data architecture of our mental health care app. First, we'll briefly recap the business requirements and then derive and explain the technical components and architecture that we aim to implement in order to meet those requirements.

Business Requirements

As already described in the solution chapter, our app consists of the three main features chat, mood tracking and journaling. Apart from that we're going to implement an onboarding dialogue to gather initial data about the user and make a first assessment of his mental health state that will allow us to select the appropriate therapeutic method and personalize the experience accordingly. Additionally, we're

going to implement a user profile, where the user can get insights based on personal statistics and update user information, such as demographics.

The key challenges we're facing will be:

- to ingest and utilize the data from the mood tracking, journaling, onboarding dialogue and user profile into the chatbot
- to add long-term memory to the chatbot to be able to remember past conversations, which is a key for providing a truly personalized experience and meaningful mental health support
- to add some kind of guardrails to the chatbot to apply appropriate therapeutic methods, identify if the user is going off-topic or having serious mental health issues that requires support from a human therapist
- to empower the chatbot to learn from conversations, gather meaningful insights and improve over time

Now that we're having this broad view, in the following, we'll break it down on a lower level in order to derive concrete technical requirements from a feature perspective:

Chatbot

The chatbot module needs to be able to:

- Apply appropriate therapeutic method / conversation style as overall framework based on mental health assessment obtained from the onboarding dialogue
- Understand which kind of mental health problem the user has based on chat conversation
- Identify if the actual problems identified in the chat conversation differs from what the user has provided in the onboarding dialogue
- Adapt the conversation style by evaluating and selecting the most appropriate method accordingly
- Learn from real therapy conversations as examples to be applied to the selected overall conversation framework
- Remember chat conversations from previous sessions
- Integrate user data from the mood tracking, journaling, onboarding dialogue and user profile to personalize conversation experience
- Identify when user is going off-topic → ask him to go back to defined scope

- Identify when users mental health situation is getting in a critical state that requires help from a real therapist → emphasize user to seek for help and provide options

The chatbot data consists of the following, structured and unstructured data:

Key	Type	Description	Example
user_id	Int	Unique identifier for users	102937282
conversation_start_date	Datetime	Date and time when the conversation has started	14.04.2023 09:48
conversation_end_date	Datetime	Date and time when the conversation has ended	14.04.2023 10:04
conversation_duration	Float (calculated)	Calculated duration of the conversation session	16.0
message_id	Int	Unique identifier for chat messages	356292124528
message_type	String	Represents whether message was sent by user or chatbot	“user_message”
message_content	String	Content of the message	“I’m feeling overwhelmed”
applied_framework	String	Therapy practice guidelines applied in this conversation	“Cognitive Behavioral Therapy”

Mood Tracking

The mood tracking module needs to:

- Store the mood tracking data provided by the user through the user interface
- Provide an endpoint for the chatbot module and user profile module

The mood tracking data consists of the following, structured data:

Key	Type	Description	Example
user_id	Int	Unique identifier for users	102937282
date	Datetime	Date and time when the entry was created	14.04.2023 09:48

Key	Type	Description	Example
mood_rating_score	Int	Numeric value describing the overall mood on a scale from 1-5	2
emotions	List	List of emotions applying to the current mood	[“tired”, “lonely”, “sad”]
activities	List	List of users’ main activities of that day	[”studying”, “social media”, “exercise”]

Journaling

The journaling module needs to:

- Store the journaling data provided by the user through the user interface
- Provide an endpoint for the chatbot module

The mood tracking data includes the following, mostly unstructured data:

Key	Type	Description	Example
user_id	Int	Unique identifier for users	102937282
date	Datetime	Date and time when the entry was created	14.04.2023 09:48
prompt_id	Int	ID of the selected prompt	482912837
journaling_prompt_message	String	Content of the selected journaling prompt	“List three positive things you’ve experienced today”
journaling_prompt_response	String	Content of the users journaling prompt response	“Finishing a big and important chapter of my master thesis, …”

Onboarding Dialogue

The journaling module needs to:

- Store the onboarding data provided by the user through the user interface

- Provide an endpoint for the chatbot module, user profile module

The onboarding dialogue includes the following, structured data:

Key	Type	Description	Example
user_id	Int	Unique identifier for users	102937282
date	Datetime	Date and time when the entry was created	14.04.2023 09:48
in_therapy_now	Boolean	Represents whether user is currently in therapy	False
in_therapy_past	Boolean	Represents whether user has been in therapy in the past	False
medication_now	Boolean	Represents whether user takes medication against mental health issues	False
medication_past	Boolean	Represents whether user takes medication against mental health issues	False
mental_health_issues	List	List of mental health issues according to user	[”stress”, ”anxiety”]
goals	List	List of goals the user wants to achieve with this app	[”reduce stress”, “prevent procrastination”]
date_of_birth	Int	Birth date of the user	24.01.1998
age	Int (calculated)	Automatically calculated age based on birth date	25
gender	String	Gender of the user	“male”
mood_rating_score	Int	Numeric value describing the overall mood on a scale from 1-5	2
emotions	List	List of emotions applying to the current mood	[“tired”, “lonely”, “sad”]
activities	List	List of users’ main activities of that day	[”studying”, “social media”, “exercise”]

User Profile

The user profile module needs to:

- Receive and store the data provided by the onboarding dialogue
- Store the data provided by the user through the user profile user interface
- Provide an endpoint for the chatbot module

The user profile includes a combination of the onboarding dialogue data and mood tracking data that can be found above. Additionally it will include some calculated data for statistics, e.g. average rating score, number of chat conversations, etc.

Technical Requirements

In the following, we'll describe the key technologies and their associated technical components that we'll be using in order to satisfy the business criteria.

Large Language Models

We've already highlighted the importance of large language models (LLMs) from business perspective in the solutions chapter. In the following we'll briefly describe the technical concept of LLMs and how we're going to integrate them into our architecture.

What are large language models?

Large language models are a type of model used in the field of Natural Language Processing (NLP). NLP is a branch of artificial intelligence that deals with the interaction between computers and humans through natural language. Due to the recent advancements, LLMs are playing an increasingly important role in that field.

One type of large language model is the transformer-based model. Examples of these include OpenAI's GPT (Generative Pre-trained Transformer) models, such as GPT-3 and GPT4. They are "transformer-based" because they use transformer architectures, which rely on self-attention mechanisms to generate text.

On a technical level, these models are designed to predict what comes next in a piece of text, which helps them generate human-like text. Furthermore, they are capable of understanding context, sentiment, and semantic meaning, which makes them powerful tools for a variety of NLP tasks, including translation, question-answering, summarization, and more.

However, it's important to note that while these models can generate remarkably human-like text, they do not understand the text in the way humans do. They don't have beliefs, opinions, or feelings, and any sentiment expressed is purely a reflection of the data they were trained on.

We’re getting back to this problem and how we’re going to address it later again.

Why do we need large language models in our app?

We’ve already covered the high potential and vast applications of these models and one of the most promising aspects is the ability to build powerful chatbots.

In our specific case, we’re going to use a chat model, which is a type of large language model that has been adapted for conversational tasks by taking a list of chat messages as input and return a chat message.

There are many chat models from different providers in the market. In our case, we’re going to use OpenAI’s GPT-4, which based on our research is the leading technology and can be accessed through OpenAI’s Chat Completions API endpoint.

What are the limitations of large language models?

As previously mentioned, there are some challenges we’re facing with large language models. The two main issues in our case are hallucinations and context limits.

Hallucinations: If the chat model doesn’t have access to specific data, it might make up answers for things it doesn’t know or have context for, which could lead to wrong answers - or in our case of a mental health chatbot - in worst case give harmful suggestions to the users.

Context limits: To give the best answers, chatbots need to understand the context of a conversation, but there's a limit to the amount of extra information a query can handle. This means they don’t have a long-term memory and thus can’t remember past conversations, which is important for having natural, ongoing conversation experiences.

However, even though these issues exist in the off-the-shelf LLMs, there are some effective measures to mitigate those risks. In our case we’re going to use embeddings stored in a vector database and fine-tuning techniques as described in the following.

Embeddings

Embeddings are a way of representing text as high-dimensional vectors (lists) of numbers, where the distance between two vectors measures their relatedness. Words or phrases with similar meanings are closer to each other than words that are semantically dissimilar.

Utilizing pre-trained embeddings can help address the "cold start" problem. Embeddings are trained on vast corpuses of text data and contain semantic

understanding of millions of words and phrases.

In our case, we’re going to use the OpenAI Embeddings API to transform large amounts of text, such as the conversation history and established therapy practices into embeddings and store them in a vector database, which can be queried by the chat model to provide additional context.

However, to make the chat model more specialized to the specific needs of mental health conversations, we will also be fine-tuning these embeddings.

Fine-Tuning

Fine-tuning is a transfer learning technique where a pre-trained model is further trained on a smaller, specific dataset. In our context, we will be fine-tuning the embeddings on a dataset of real therapy conversations and information about therapy practices. This allows us to leverage the vast knowledge included in the pre-trained embeddings while adapting it to the specificities of mental health therapy.

The fine-tuning process allows the model to learn from the specific context and language used in therapy conversations without the need to provide examples in the prompt. It helps the model understand the subtleties of therapeutic language and the various strategies employed by therapists. For instance, the model can learn to reflect and validate feelings, ask open-ended questions, and use other therapeutic techniques. This makes the chatbot more equipped to handle conversations pertaining to mental health.

In our case we’re going to use OpenAI’s integrated fine-tuning solution.

Vector Database

A vector database is a data storage system that is designed to handle vector data efficiently. In the context of our chat model, the vector database will store the embeddings of all the conversation history of the user with the chatbot to address the problem of context limits. This allows the chatbot to access and consider past conversations when generating responses, thereby enabling a more coherent and context-aware conversation.

The vector database will also store the embeddings of therapy practice information. This allows the chatbot to retrieve and apply relevant therapeutic strategies based on the user's inputs. For example, if a user is displaying signs of cognitive distortion, the chatbot can retrieve and apply cognitive behavioral therapy (CBT) techniques that are relevant to the situation.

So basically, the vector database serves as the memory of the chatbot, enabling it to remember past interactions and apply learned therapeutic strategies effectively.

In our case we’re going to use Pinecone, which is a managed vector database service especially in context of large language model applications a leading provider.

LangChain

LangChain is the most important python library in the LLM chat model space that - as the name suggests - provides a framework to integrate LLMs with other tools and services enabling them to interact with their environment.

In case of our mental health app, the usage of the LangChain framework allows us to efficiently connect the chat model with the other modules and data sources described above.

The LangChain framework provides a lot of components already and expands rapidly.

Following, three of most important components that will be very useful for our use case:

Chains

Chains is an incredibly generic concept which returns to a sequence of modular components (or other chains) combined in a particular way to accomplish a common use case.

The most commonly used type of chain is an LLMChain, which combines a PromptTemplate, a Model, and Guardrails to take user input, format it accordingly, pass it to the model and get a response, and then validate and fix (if necessary) the model output.

One use case of a chain in our case would be to instruct the model to apply a certain therapy practice in a conversation based on the outcome of the assessment from the onboarding dialogue and to store a summary of the conversation as embedding to the vector database.

Agents

Some applications need more than a pre-set chain of calls to LLMs or other tools; they might require a chain that changes based on user input. Here, an "agent" with a set of tools decides which tool to use based on the input.

The key components in this agent architecture are:

- **Agent**: Handles app logic, takes user input and prior steps, and returns an AgentAction or AgentFinish.
- **Tools**: Actions the agent can take, chosen based on the agent's purpose.
- **Toolkits**: Groups of tools for specific use cases.
- **Agent Executor**: Runs the agent iteratively with a list of tools until a stopping point.

In our scenario, an agent could autonomously decide the suitable therapy method based on defined criteria. It could assess if the user's behavior differs from initial onboarding information and choose an appropriate therapy technique based on its observations.

Memory

LangChain offers a variety of memory types, functions, and classes, allowing the integration of both short-term and long-term memory into the application.

In addition to the conversation storage and retrieval mentioned earlier, LangChain also offers a conversation summary memory type, which could be used in our case to generate and store summaries at the end of a conversation.