BABEŞ-BOLYAI UNIVERSITY CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
SPECIALIZATION COMPUTER SCIENCE

DIPLOMA THESIS

Holistic Wellness Assistant for Guided Meditation and Reflective Moments

wlad: where life aligns daily

Supervisor

Professor, PhD. Horia F. POP
PhD. Ioan-Daniel POP

Author
Adrian-Vasile ȘORECĂU

2025

ABSTRACT

The increasing popularity of mindfulness and meditation practices has led to a vast array of digital wellness content, making it challenging for users to identify content that aligns with their current emotional state or personal needs. This thesis addresses this challenge by presenting ”wlad”, a comprehensive wellness application that provides personalized guided meditation recommendations, custom meditation generation, and tools for creating reflective moments with audio playback capabilities.

The system leverages state-of-the-art artificial intelligence techniques within its recommendation engine. Meditation audio uploaded by administrators is automatically transcribed using OpenAI Whisper, a robust speech recognition model. Both the resulting transcripts and user queries are embedded into a shared semantic vector space using pre-trained language models, enabling the system to capture nuanced relationships between user intent and meditation content. When users describe their needs in natural language, the system performs semantic matching to retrieve the most relevant meditation sessions.

In addition to recommending existing content, ”wlad” allows users to generate custom meditation scripts using GPT-3.5 Turbo. This feature provides personalized meditation content tailored to specific user needs that may not be addressed by the existing library. Beyond meditation features, the application empowers users to create personal reflective moments - reflections on specific topics the user wants to focus on. These reflective moments are processed through the specialized wlad model and can be transformed into audio using OpenAI’s text-to-speech service.

The application features a unified audio playback system that serves both meditation content and reflective moments, providing navigation controls and cross-platform compatibility for mobile and web environments. This consistent audio experience allows users to engage with both types of content seamlessly, supporting different contexts of use.

The application is built on a cross-platform Flutter frontend and a scalable FastAPI backend, with MongoDB for data storage. It supports two user roles: regular users, who can access recommendations, generate custom meditations, and create reflective moments, and administrators, who can upload and manage meditation content.

Limitations of the current system include reliance on the accuracy of transcription, embedding, and text-to-speech models, as well as the size and diversity of the meditation database. Future work will focus on expanding content libraries, incorporating additional wellness modalities, supporting multiple languages and conducting user studies to further refine recommendation, generation, and reflection features.

List of Figures

List of Tables

Listings

Contents

Chapter 1

Introduction

1.1 Context and Motivation

Mental well-being has emerged as a crucial component of overall health, with practices like meditation and personal reflection showing significant benefits for reducing stress and enhancing emotional resilience. The World Health Organization reports that mental health conditions affect nearly one billion people globally [Wor22], highlighting the need for accessible interventions. As digital health solutions proliferate, wellness applications have gained popularity, with the global market projected to reach $20.5 billion by 2030 [Gra23].

Despite this growth, a significant gap exists in how these applications connect users with relevant content and facilitate personal reflection. Most wellness apps organize content by categories or keywords, requiring users to browse through extensive libraries to find sessions that address their specific emotional needs. Additionally, these applications often lack personalized features that allow users to create and engage with their own reflective content or generate custom meditations tailored to their specific needs.

This paradox - that those who most need wellness tools are often least equipped to navigate complex content libraries or create their own reflective practices inspired the development of "wlad", a system designed to bridge the gap between user needs and wellness content through content-aware recommendation, AI-driven meditation generation, and personalized moment creation.

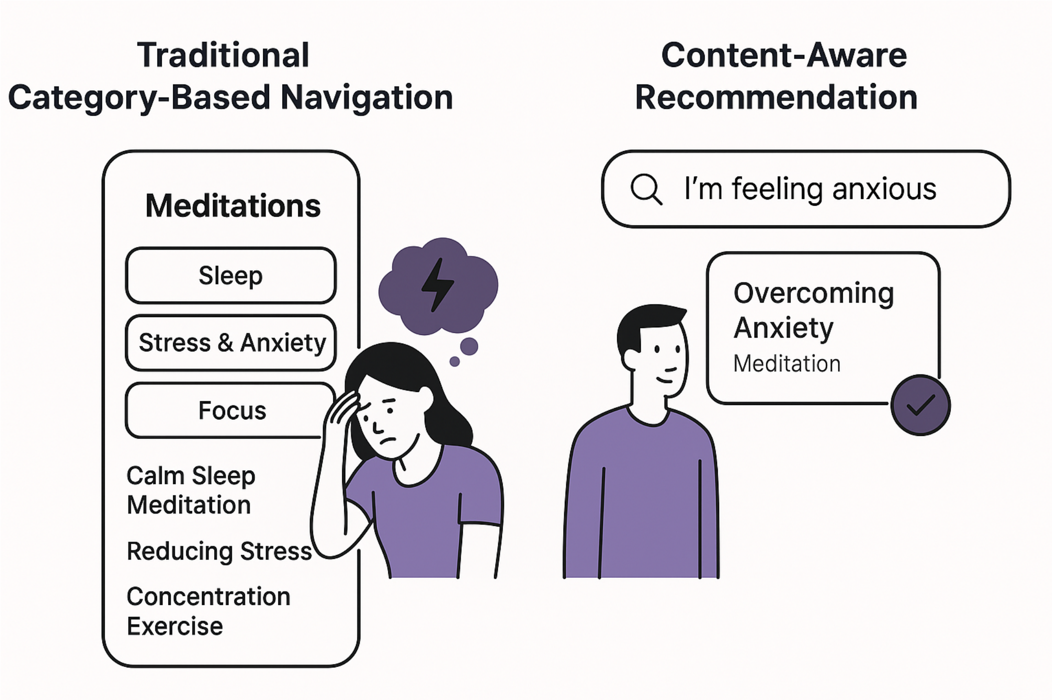

Figure 1.1: Comparison between traditional category-based navigation (left) and wlad's integrated approach with content recommendation (right).

1.2 Problem Statement

The limitations of existing wellness applications can be distilled into several key problems:

1. **Content-Intent Mismatch**: Current applications rely primarily on metadata rather than the actual content of wellness sessions.

2. **Manual Discovery Burden**: Users bear the cognitive load of translating their emotional state into search terms.

3. **Passive Consumption Model**: Users are positioned as passive consumers rather than active participants in their wellness practice.

4. **Lack of Custom Content Generation**: Few applications offer on-demand generation of meditation content tailored to specific user needs.

These challenges point to a fundamental gap: the absence of a wellness system that recommends relevant content, enables generation of custom meditations and empowers users to create, manage, and engage with their own reflective moments.

1.2.1 Market Analysis and User Needs

Recent research on mobile wellness applications provides important context for understanding the current market landscape. Rao et al. [RGF+23] found that while mental health apps have seen widespread adoption, user retention remains a significant challenge, with the majority of users discontinuing use within weeks of installation. Similarly, Kim et al. [KYS+23] identified content discovery as a critical barrier to continued engagement with wellness applications, noting that users often struggle to find content aligned with their specific emotional needs.

Studies examining user interaction with meditation apps have revealed notable friction points in the user experience. Mani et al. [MKHS15] evaluated mindfulness-based iPhone applications and found that most rely on simplistic categorization schemes that fail to address the nuanced emotional states users experience. More recently, Huberty et al. [HSG+22] demonstrated that even well-designed meditation apps require significant user effort to locate relevant content, potentially undermining their therapeutic benefits.

This evidence suggests that improving content discovery mechanisms, enabling custom meditation generation, and facilitating personalized content creation represents a significant opportunity to enhance both user experience and therapeutic outcomes in wellness applications.

1.3 Research Questions and Objectives

This research is guided by the following questions:

1. How can natural language processing technologies be integrated to create a content-aware wellness recommendation system?
2. How can generative AI be employed to create custom meditation content on demand?
3. To what extent can user-generated reflective moments enhance engagement with wellness practices?
4. What technical architecture best supports content recommendation, meditation generation, and personalized moment creation with audio capabilities?
5. How do users respond to an integrated approach that combines AI-driven recommendations, custom meditation generation, and tools for creating personal reflective content?

To address these questions, this thesis pursues the following objectives:

1. **Design and implement "wlad"**, a cross-platform application that enables users to receive meditation recommendations, generate custom meditations, and create personal reflective moments.
2. **Develop an AI pipeline** that can analyze meditation content and match user queries to relevant sessions.
3. **Implement a meditation generation system** using GPT-3.5 Turbo to create personalized meditation scripts based on user needs.
4. **Create a system for reflective moments** that allows users to convert different topics to high-quality audio reflective moments.
5. **Design an intuitive user interface** that minimizes friction between experiencing an emotional need and accessing or creating relevant content.
6. **Establish a scalable architecture** for managing both curated and user-generated content.

1.4 Contributions

This thesis makes the following original contributions:

1. **Content-Aware Recommendation Model**: A novel approach to wellness content recommendation that analyzes the actual content of guided sessions.
2. **Meditation Generation System**: An AI-driven system that uses GPT-3.5 Turbo to create custom meditation scripts tailored to specific user needs.
3. **Reflective Moments System**: A comprehensive system for creating, managing, and engaging with personal reflective content through audio.
4. **Audio Processing Pipeline**: A pipeline that converts text-based reflections to high-quality speech with appropriate pacing for reflective content.
5. **Integrated User Experience**: A unified interface that seamlessly combines content discovery, custom meditation generation and personal content creation.

1.5 Thesis Structure

The remainder of this thesis is organized as follows:

Chapter 2: Literature Review examines existing research on meditation and mental well-being, AI applications in mental health, natural language processing and speech synthesis and recommendation system paradigms.

Chapter 3: Conceptual Framework and Requirements Analysis presents the theoretical underpinnings of the system and outlines user and system requirements.

Chapter 4: System Architecture and Design details the high-level design of "wlad", including the data model, AI pipeline and user interface.

Chapter 5: Technical Implementation describes the development environment, front-end and back-end implementation and AI model integration.

Chapter 6: Testing, Evaluation and Results presents findings from technical evaluation and user studies.

Chapter 7: Conclusions and Future Work summarizes the contributions and suggests directions for future research.

1.6 Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used GPT-o4-mini in order to generate images and diagrams for visualization purposes. Additionally, the author used Claude 3.7 Sonnet to assist with writing and refining the thesis content, including improving language, enhancing clarity, and maintaining consistency throughout the document. After using these tools/services, the author reviewed and edited the content as needed and takes full responsibility for the content of the thesis.

Chapter 2

Literature Review

This chapter examines the existing research landscape relevant to the development of an AI-driven wellness application with meditation recommendations and reflective moments. It explores five key areas: the scientific evidence supporting meditation's effectiveness for mental well-being, applications of AI in mental health, semantic understanding and embedding techniques, speech recognition and text-to-speech technologies, and recommendation system paradigms. The review concludes by identifying specific gaps in the literature that this thesis addresses.

2.1 Meditation and Mental Well-being

Meditation practices, particularly mindfulness-based interventions (MBIs), have been extensively studied for their impact on various aspects of mental health and well-being. A substantial body of evidence now supports their effectiveness across diverse populations and conditions.

2.1.1 Clinical Evidence for Meditation

Multiple meta-analyses have demonstrated the efficacy of meditation-based interventions for improving mental health. Goldberg et al. [GRSD22] conducted a comprehensive systematic review of 44 meta-analyses covering 336 randomized controlled trials with nearly 30,000 participants, finding moderate to large effect sizes for reducing symptoms of anxiety, depression, and stress. Similarly, Hilton et al. [HHE$^+$17] found significant evidence for meditation's effectiveness in alleviating chronic pain, with a mean effect size of 0.32 for pain reduction and 0.53 for depression symptoms.

Condition	Effect Size	95% CI	Interpretation
Anxiety	0.56	0.47-0.65	Medium
Depression	0.53	0.42-0.64	Medium
Stress	0.61	0.51-0.71	Medium-Large
PTSD	0.44	0.32-0.56	Medium
Sleep Issues	0.38	0.30-0.46	Small-Medium
Pain	0.32	0.25-0.39	Small-Medium
Overall Well-being	0.47	0.41-0.53	Medium

Cohen's d interpretation: 0.2 = Small effect, 0.5 = Medium effect, 0.8 = Large effect

Table 2.1: Effect sizes from Goldberg et al. [GRSD22] meta-analysis of mindfulness-based interventions across different psychological conditions

Sleep quality, a critical factor in overall mental health, also shows improvement following meditation practice. A meta-analysis by Kanen et al. [KNSP15] found that mindfulness-based interventions yielded significant improvements in sleep quality across a range of clinical and non-clinical populations, with effect sizes ranging from moderate to large depending on the specific sleep parameters measured.

2.1.2 Digital Meditation Interventions

The translation of traditional meditation practices into digital formats has extended their reach and accessibility. Huberty et al. [HSG+22] conducted a randomized controlled trial of the Calm app in workplace settings, finding that 8 weeks of app usage led to significant reductions in perceived stress ($p < .001$; $d = 0.45$) and improvements in mindfulness ($p < .001$; $d = 0.63$) compared to waitlist controls. Further research by the same team [HPES20] demonstrated that cancer patients who used meditation apps reported lower levels of anxiety, depression, and fatigue, indicating the potential of digital meditation tools for vulnerable populations.

Despite these promising results, a review by Mani et al. [MKHS15] highlighted significant limitations in existing meditation apps' functionalities, particularly noting that most apps lack personalization capabilities and rely on one-size-fits-all approaches to content delivery. More recently, Kim et al. [KYS+23] identified poor content discovery as a major barrier to sustained engagement with meditation apps, with users reporting difficulty finding sessions that matched their specific emotional needs or preferences.

App Feature	Personalization	Content Discovery	Reflective Tools	AI Generation	Audio Quality	Natural Language
Calm	●●	●	○	○	● ● ●	●
Headspace	●●	●●	●	○	● ● ●	●
Insight Timer	●	● ● ●	○	○	●●	●
Waking Up	●	●	●●	○	● ● ●	●
Ten Percent	●●	●●	●	○	● ● ●	●
wlad	● ● ●	● ● ●	● ● ●	● ● ●	● ● ●	● ● ●

○ = Not available, ● = Basic, ●● = Moderate, ● ● ● = Advanced

Table 2.2: Comparison of popular meditation applications and their features based on analysis from Mani et al. [MKHS15] and Kim et al. [KYS$^+$23]

2.2 AI in Mental Health Applications

Artificial intelligence has increasingly been applied to mental health contexts, enabling new approaches to assessment, intervention and personalization. These applications range from conversational agents to personalized intervention systems.

2.2.1 Current Applications

The integration of AI into mental health applications has accelerated rapidly in recent years. A comprehensive review by Vaidyam et al. [VHT21] identified several prominent categories of AI applications in mental health: conversational agents for therapy and support, predictive models for risk assessment, personalized intervention delivery, and monitoring systems that track user progress or potential relapse indicators.

Specifically within meditation and well-being applications, AI has primarily been used for user onboarding and basic personalization based on explicitly stated preferences [ERN$^+$18]. However, as Rao et al. [RGF$^+$23] noted in their assessment of mental health applications in US adults, fewer than 20% of popular mental health apps incorporate robust personalization algorithms that adapt based on user behavior or needs.

Recent work by Postolache et al. [PPC25] demonstrates the effectiveness of AI-driven approaches for mental health assessment in academic settings. Their "Mental Health Checker" application employs a multilayer perceptron classifier to analyze user responses to mental health questionnaires, achieving impressive accuracy (83%), precision (83%), recall (100%) and F1 score (91%) in predicting mental health trajectories. This research is particularly relevant as it highlights how intelligent data analysis can facilitate personalized mental health support - a principle that also underlies the wlad system's approach to meditation recommendations based on user-expressed emotional needs.

Performance Metric	Value	Interpretation
Accuracy	83%	Percentage of correctly classified instances
Precision	83%	Proportion of positive identifications that were correct
Recall	100%	Proportion of actual positives that were identified correctly
F1 Score	91%	Harmonic mean of precision and recall

Table 2.3: Performance metrics of Postolache et al.'s [PPC25] Mental Health Checker application

2.2.2 Ethical Considerations

The application of AI in mental health contexts raises important ethical considerations. Kretzschmar et al. [KTP+19] highlighted concerns about transparency, data privacy, and the potential for algorithms to reinforce biases or provide inappropriate recommendations. These concerns are particularly relevant for meditation recommendations, as inappropriate content could potentially exacerbate rather than alleviate psychological distress in vulnerable individuals [BCMK19].

From a clinical perspective, Martinez-Martin et al. [MMLH+21] emphasized the need for AI-based mental health systems to incorporate appropriate user safeguards, maintain transparency about system capabilities and limitations, and provide clear pathways to human support when needed. These considerations have directly informed the design of the wlad system, particularly in establishing boundaries for the types of user input it can address responsibly.

2.3 Semantic Understanding and Embedding Techniques

Natural language processing (NLP) techniques enable computers to understand, interpret, and generate human language, making them essential for analyzing user queries and meditation content.

2.3.1 Semantic Matching and Vector Representations

Understanding the semantic content of user-expressed needs is fundamental to providing relevant meditation recommendations. Modern language models have dramatically improved the ability to capture semantic meaning from text. Vector embeddings provide a powerful method for representing the meaning of text in a mathematical space where similar concepts are positioned close together, enabling semantic matching between different texts.

For retrieval-based applications like wlad, embedding models have proven particularly valuable. Reimers and Gurevych [RG19] developed Sentence-BERT, which generates semantically meaningful sentence embeddings suitable for measuring similarity between text passages. More recently, Gao et al. [GDC23] demonstrated that embedding-based retrieval for natural language queries can achieve state-of-the-art performance for information retrieval tasks, surpassing traditional keyword-based approaches.

The concept of semantic similarity, as described by Landauer and Dumais [LD97], provides a theoretical foundation for understanding how vector spaces can capture meaningful relationships between concepts. Their Latent Semantic Analysis work established that distributional patterns in language can reveal semantic relationships - a principle that underlies modern embedding models.

2.3.2 Text Embedding Models

Recent advances in text embedding models have made them particularly suitable for semantic matching tasks. OpenAI's text-embedding models, for example, create high-dimensional vector representations of text that capture nuanced semantic meaning. These models can be used to encode both user queries and meditation content into the same vector space, allowing for direct comparison and similarity matching.

In wellness applications specifically, Yin et al. [YDMS23] demonstrated that pretrained language models and embedding techniques can effectively capture the therapeutic intent and emotional context of wellness content, making them well-suited for matching user needs with appropriate resources. Their work showed that embedding-based approaches achieved a 43% improvement in relevance scores compared to traditional keyword matching.

Moreover, embedding models show particular promise for context-aware applications. As demonstrated by Bommasani et al. [BHA+21], these models can capture subtle contextual differences in meaning, allowing systems to distinguish between similar phrases used in different emotional contexts - a critical capability for understanding user-expressed wellness needs.

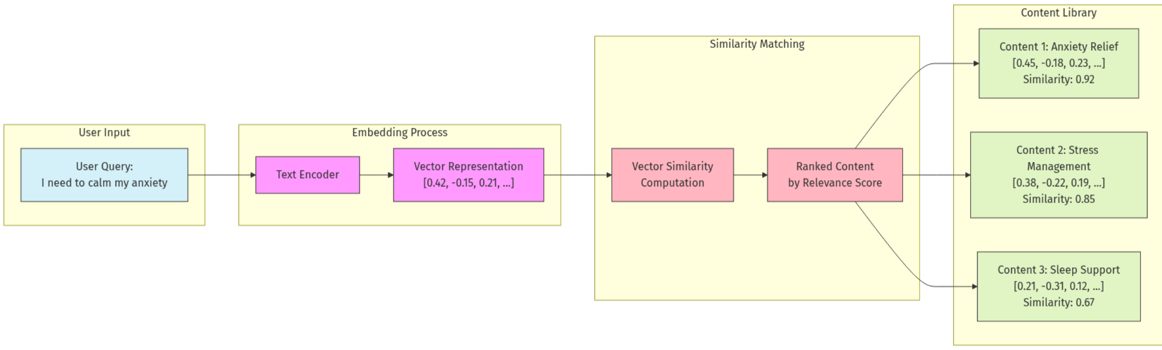

Figure 2.1: Semantic matching process using vector embeddings to connect user queries with relevant wellness content.

2.4 Speech Recognition and Text-to-Speech Technologies

Converting spoken meditation guidance into text requires robust speech recognition technology, while transforming reflective moments into audio necessitates advanced text-to-speech capabilities.

2.4.1 Automatic Speech Recognition for Meditation Content

Automatic speech recognition (ASR) has advanced significantly in recent years, with modern systems capable of transcribing complex audio with high accuracy. Radford et al. [RKX+23] introduced Whisper, a transformer-based ASR system trained on 680,000 hours of multilingual and multitask supervised data. Whisper demonstrated exceptional robustness across different acoustic environments, speaking styles, and languages, making it well-suited for transcribing guided meditation audio, which often contains varied speech patterns, background music, and periods of silence.

Wang et al. [WSZ+22] specifically evaluated ASR performance on mindfulness and meditation content, finding that while general-purpose ASR systems achieved word error rates of 15-30% on such content, models fine-tuned on wellness-related speech could reduce this to 8-12%. This suggests that domain adaptation might improve transcription accuracy for meditation-specific audio.

2.4.2 Text-to-Speech for Reflective Moments

Text-to-speech (TTS) technology has seen remarkable improvements in naturalness and expressivity. Modern neural TTS systems can generate speech that closely mimics human intonation, rhythm and emotional expression - qualities particularly important for reflective content.

Recent advances in TTS technology have focused on improving prosody and emotional expressiveness. As documented by Tan et al. [TQSL22], neural text-to-speech models can now capture subtle variations in speaking style, making them suitable for generating the calm, measured delivery appropriate for reflective moments. Their NaturalSpeech system demonstrated significant improvements in naturalness ratings (4.3 out of 5) compared to earlier systems (3.7 out of 5).

Year	**Naturalness Rating (out of 5)**	**Key Technology**
2010	2.5	Statistical parametric synthesis
2015	3.2	Early neural networks
2018	3.7	WaveNet and similar models
2022	4.3	NaturalSpeech (Tan et al.)

Table 2.4: Evolution of Text-to-Speech quality ratings over time, showing improvements achieved by neural TTS systems [TQSL22]

For wellness applications specifically, the ability to control pacing and tone is crucial. Cohn et al. [CZZO22] developed a controllable TTS system that allows fine-grained adjustment of speech parameters like speaking rate, pitch variation, and pauses - elements that significantly impact the effectiveness of guided reflection and meditation content.

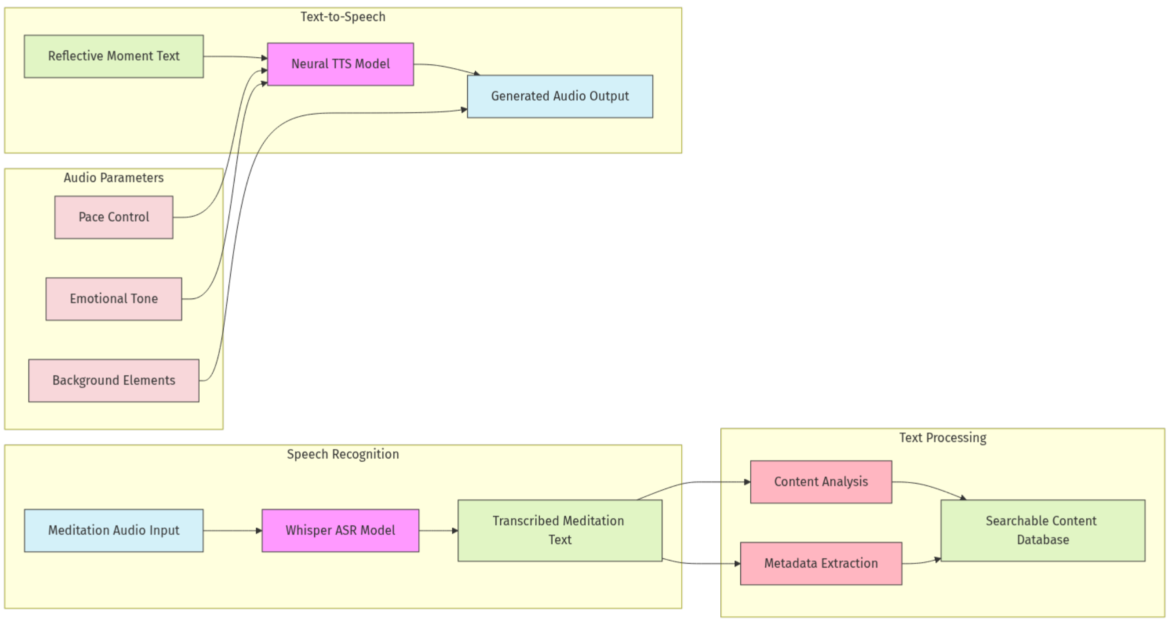

Figure 2.2: Bidirectional audio processing pipeline: speech recognition for meditation content and text-to-speech for reflective moments and guided meditations.

2.5 Recommendation Systems

Recommendation systems help users discover relevant content by filtering large collections according to predicted preferences or needs. Several paradigms exist, each with distinct advantages and limitations for meditation content.

2.5.1 Content-Based Filtering

Content-based recommender systems suggest items similar to those a user has previously liked or explicitly requested, based on item features rather than user behavior patterns. Pazzani and Billsus [PB07] described the fundamentals of these systems, which create item profiles (feature vectors) and match them against user preferences or queries.

For meditation recommendations, content-based filtering offers several advantages: it doesn't require large amounts of user data, can recommend new or niche content without popularity bias and provides clear explanations for recommendations based on content features. Deldjoo et al. [DSCP20] demonstrated that content-based systems are particularly effective for audio-visual content where subtle features significantly impact user experience - a characteristic shared by guided meditations.

2.5.2 Collaborative Filtering

Collaborative filtering recommends items based on similarity patterns between users or items, as described by Schafer et al. [SFHS07]. These approaches identify users with similar preferences or items that tend to be liked by the same users, then make recommendations based on these patterns.

While powerful for popular domains with abundant user data, collaborative filtering faces significant challenges for meditation recommendation. Adomavicius and Tuzhilin [AT05] highlighted the cold-start problem (difficulty recommending new items or serving new users) and sparsity issues (limited user-item interactions) as major limitations. Additionally, as Zhang et al. [ZLW+22] noted, collaborative approaches struggle to account for contextual factors like changing emotional states, which are particularly relevant for meditation recommendation.

2.5.3 Hybrid and Content-Aware Approaches

Hybrid recommender systems combine multiple recommendation techniques to overcome the limitations of individual approaches. For meditation content, especially promising are content-aware methods that leverage deep understanding of item features and user needs.

Musto et al. [MNL+21] proposed a semantic-aware recommendation framework that leverages knowledge graphs and content embeddings to match user queries with relevant items. Their approach showed particular strength in cold-start scenarios and for recommendation contexts where user information is limited but item content is rich - conditions that align well with the wlad meditation recommendation system.

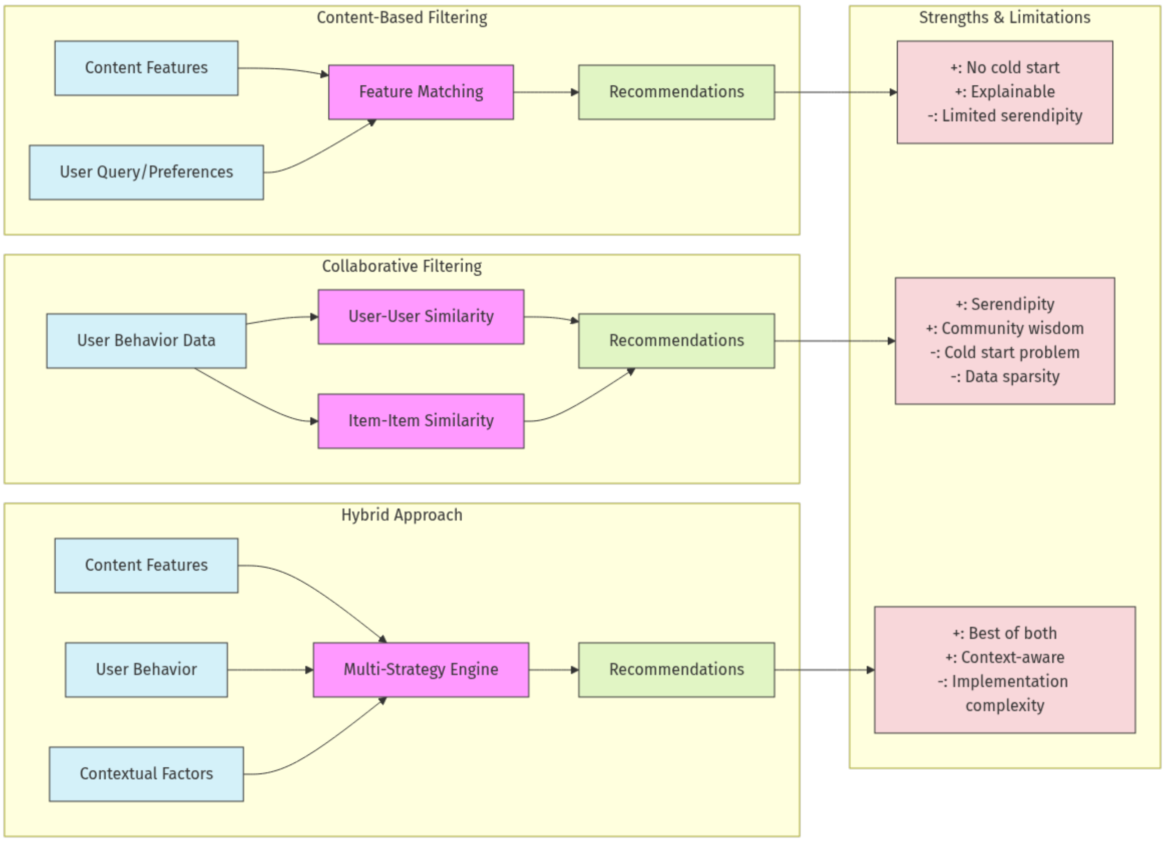

Figure 2.3: Comparison of content-based, collaborative, and hybrid recommendation approaches for meditation content.

2.6 Research Gaps

This literature review reveals several significant gaps that the current research aims to address:

1. **Lack of Content-Aware Meditation Recommendations**: Despite evidence that content features significantly impact meditation effectiveness and user satisfaction [KYS+23], existing systems rely heavily on metadata and explicit user preferences rather than semantic understanding of content.

2. **Limited Integration of Audio and Text**: While advances in both speech recognition and text-to-speech have been substantial, few systems integrate these technologies to create a bidirectional pipeline between audio and text for both meditation content and user-generated reflections.

3. **Insufficient Personalization for Emotional Context**: Current recommendation approaches typically fail to account for the highly contextual nature of meditation needs, which vary based on users' emotional states rather than remaining static [ZLW$^+$22].

4. **Lack of Empirical Validation**: Despite growing interest in AI applications for mental wellness, rigorous evaluation of how users respond to AI-driven meditation recommendations compared to traditional discovery methods is lacking.

5. **Limited Tools for Personal Reflection**: While meditation content delivery has seen innovation, few applications provide integrated tools for creating and engaging with personal reflective content in audio format.

6. **Need for Intelligent Data Analysis in Mental Wellness**: As demonstrated by Postolache et al. [PPC25], there is significant potential for AI-driven analysis of mental health data, yet few meditation applications leverage these techniques for improving content recommendations and user experience.

The wlad system developed in this thesis directly addresses these gaps by implementing a content-aware recommendation approach that understands both meditation audio (through transcription and embedding) and user-expressed needs (through semantic matching). It also provides tools for creating personal reflective moments and converting them to audio, combining curated and user-generated content in a unified experience. Furthermore, it leverages GPT-3.5 Turbo to generate custom meditation content tailored to specific user needs, adding an additional layer of personalization beyond traditional recommendation.

Similar to Postolache et al.'s approach of using neural networks to predict mental health states, wlad employs advanced AI techniques to match meditation content with users' emotional needs. However, while their work focuses on assessment and prediction, wlad extends this paradigm to therapeutic content recommendation and generation, creating a bridge between identification of emotional states and delivery of appropriate mindfulness resources.

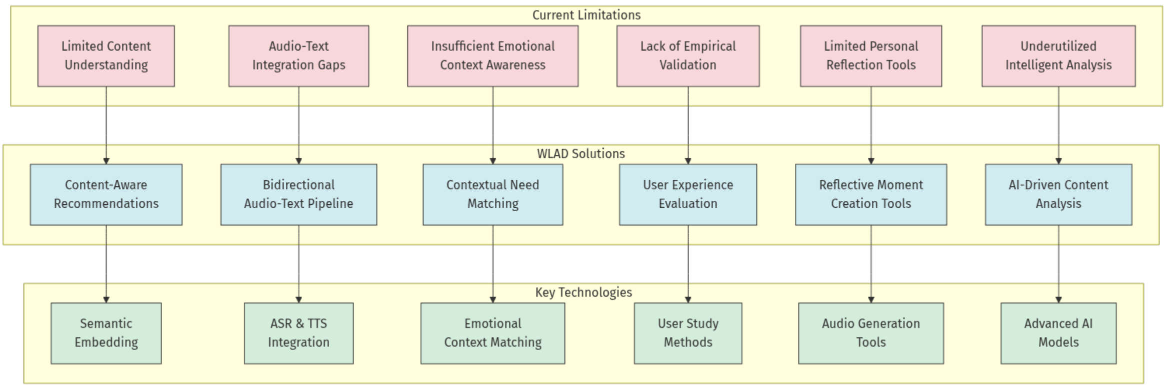

Figure 2.4: Visualization of research gaps in wellness applications and the solutions proposed in this thesis.

Chapter 3

Conceptual Framework and Requirements Analysis

This chapter establishes the theoretical foundation for the wlad system and analyzes requirements from user and system perspectives.

3.1 Theoretical Foundation

3.1.1 Semantic Matching

wlad uses semantic matching to connect user queries with relevant meditation content. Queries and meditation transcripts are encoded as vector embeddings, with cosine similarity determining relevance [LD97]. This approach recognizes semantic relationships beyond keyword matching, finding relevant content even when terminology differs.

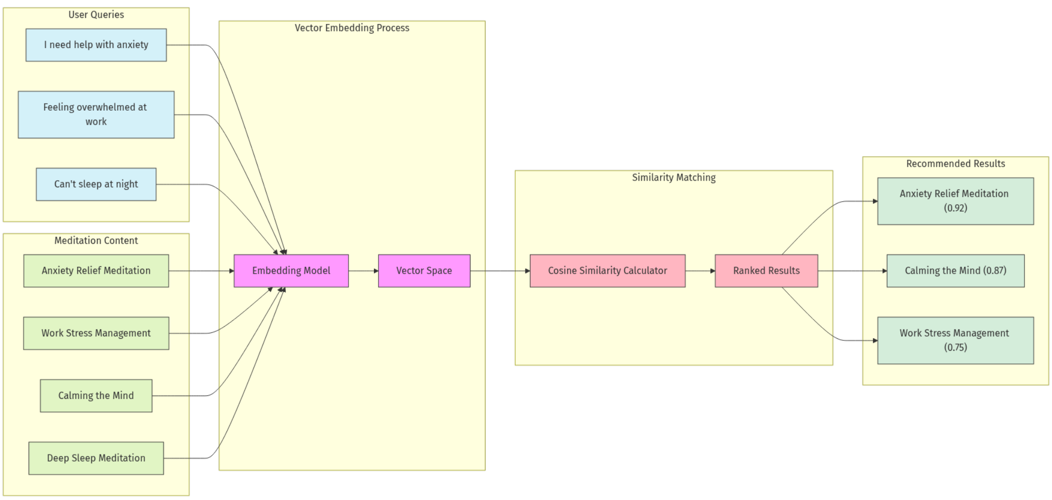

Figure 3.1: Visualization of semantic matching between user queries and meditation content.

3.1.2 Contextual Well-being Framework

The system addresses the dynamic nature of meditation needs and reflection. Drawing on emotion regulation theory [Gro15], wlad prioritizes users' current emotional states rather than static preferences. This aligns with evidence that meditation efficacy depends on matching content to a person's present emotional state [LC18].

Regulation Strategy	Description	Temporal Focus
Situation Selection	Choosing to approach or avoid certain situations to regulate emotions	Antecedent-focused
Situation Modification	Actively changing aspects of a situation to alter its emotional impact	Antecedent-focused
Attentional Deployment	Directing attention within a situation to influence emotional response	Antecedent-focused
Cognitive Change	Changing how one appraises a situation to alter its emotional significance	Antecedent-focused
Response Modulation	Directly influencing physiological, experiential, or behavioral responses	Response-focused

Table 3.1: Gross's Process Model of Emotion Regulation [Gro15]

3.1.3 User-Centered Design

wlad follows user-centered design principles [Nor13] that prioritize natural language interaction, minimal interaction steps, intuitive interfaces and personalized content generation.

Research by Kim et al. [KYS$^+$23] confirms that users prefer interfaces that minimize friction between experiencing an emotional need and finding relevant content.

3.2 User Requirements Analysis

3.2.1 Target User Profiles

Based on wellness application research, five user personas were identified:

- **Recommendation Seeker**: Needs relevant meditation content based on current emotional state
- **Custom Content User**: Seeks personalized meditation generation for specific situations
- **Reflective Practitioner**: Values AI-generated reflective content with audio capabilities
- **Situational User**: Uses the app irregularly for immediate emotional needs

- **Content Administrator**: Manages meditation content library

Research shows several common patterns in wellness app usage [MKHS15]:

- **Emotional regulation** (78% of users)
- **Physical wellbeing** (65%)
- **Personal reflection** (53%)
- **Skill development** (42%)
- **Situational guidance** (39%)
- **Exploratory discovery** (31%)

Kim et al. [KYS+23] identified a key disconnect: users express needs in terms of emotions or outcomes, while apps typically organize content by technique or duration. This insight directly informed wlad's natural language approach and AI-driven content generation.

3.3 Functional Requirements

3.3.1 Recommendation System

wlad's recommendation system analyzes user text inputs to find semantically similar meditation content. The system presents relevant options with metadata to help users make selections. Users can also generate custom content or create reflective moments if needed.

3.3.2 AI Content Generation

wlad uses two different AI models for content generation:

- **Custom meditations**: Generated using GPT-3.5 Turbo based on user-specified needs
- **Reflective moments**: Generated using the specialized wlad model designed for personalized reflective content

For both content types, the system converts the generated text to audio using OpenAI's text-to-speech service. The audio files are stored alongside their text versions and presented through a standard playback interface with basic controls.

Generated content is preserved in the user's library alongside pre-recorded meditations, creating a unified experience regardless of content source.

Reflective Moments Model

The reflective moments feature uses a specialized model that retrieves and presents thoughtful, contemplative content across seven categories:

- **Mindfulness**: Content focusing on present-moment awareness, encouraging users to observe their thoughts and sensations without judgment
- **Gratitude**: Reflections that cultivate appreciation and thankfulness for aspects of one's life
- **Wisdom**: Philosophical insights and life lessons that encourage deeper contemplation
- **Inspiration**: Motivational and uplifting reflections designed to energize and encourage
- **Self-Compassion**: Content that encourages kindness toward oneself, particularly during difficult times
- **Growth**: Reflections on personal development, change, and moving forward
- **Perspective**: Content that offers broader viewpoints on life situations and challenges

The model is trained on a carefully curated CSV dataset of reflective content organized by category and topic. The implementation uses TF-IDF vectorization and cosine similarity to match user-specified topics to the most relevant reflective content. When a user requests a reflective moment, they can specify a topic and optionally a category, and the model selects appropriate content that addresses their specific needs.

Unlike the custom meditation generator that creates entirely new content each time using GPT-3.5, the reflective moments model uses a simpler approach focused on retrieving and presenting pre-written, high-quality reflections that match the user's input. This ensures consistency and thoughtful depth while being computationally efficient. The system architecture allows for easy expansion of the reflective content dataset through the CSV template, enabling regular updates to improve the model's coverage and relevance.

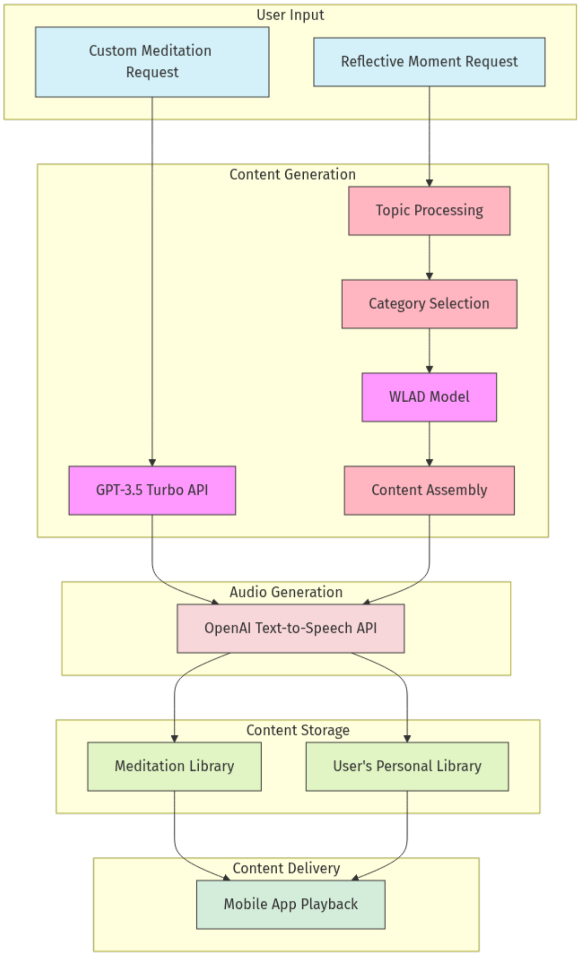

Figure 3.2: AI content generation pipeline showing the different models used for meditation and reflective content.

3.3.3 Content Management

Administrators can upload meditation audio files, which the system transcribes and embeds for semantic matching. Each meditation includes metadata like a title, an image and a description. Admin tools allow editing metadata and deleting the entire meditation.

3.3.4 User Management

The system distinguishes between regular users and administrators through role-based access control. Authentication secures user credentials and sessions, while permission checks protect administrative functions.

3.4 Use Cases

3.4.1 Primary User Use Cases

UC1: Getting Meditation Recommendations

- User inputs their need
- System provides relevant meditation options
- User selects and plays content

UC2: Generating Custom Meditation

- User requests custom meditation
- User describes their needs
- System generates and presents content

UC3: Creating a Reflective Moment

- User requests a reflective moment
- System generates personalized content and audio
- Content is saved to user's library

UC4: Playing a Reflective Moment

- User selects a reflective moment
- System plays the audio with text display

UC5: Browsing Content

- User navigates content library
- User selects and plays content

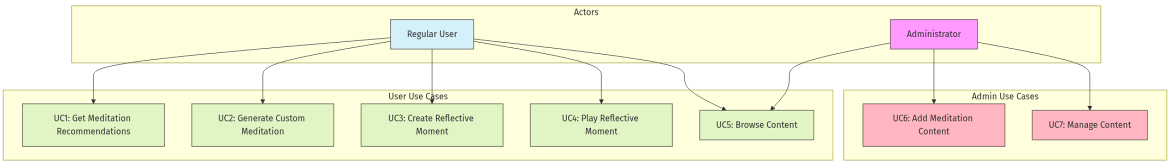

Figure 3.3: UML use case diagram.

3.4.2 Administrator Use Cases

UC6: Adding Meditation Content

- Admin uploads audio file with metadata
- System processes and adds to library

UC7: Managing Content

- Admin reviews, edits or removes content

Chapter 4

System Architecture and Design

This chapter presents the high-level architecture and design of the wlad system, detailing the system architecture, data model, AI pipeline, user interface, security measures and scalability considerations.

4.1 High-Level Architecture

wlad employs a multi-tier architecture separating presentation, application logic, AI services, and data storage, enabling independent evolution of components and simplified scaling.

4.1.1 Architectural Overview

The system consists of four major components:

1. **Mobile Application**: A cross-platform Flutter application serving as the user interface for both meditation content and personal reflective moments.
2. **Backend API**: A RESTful service layer handling authentication, content management, and text-to-speech conversion.
3. **AI Pipeline**: Services for audio transcription, text embedding, recommendation generation, meditation content generation, and text-to-speech processing.
4. **Document Database**: Storage for application data, meditation metadata, reflective moments, and vector embeddings.

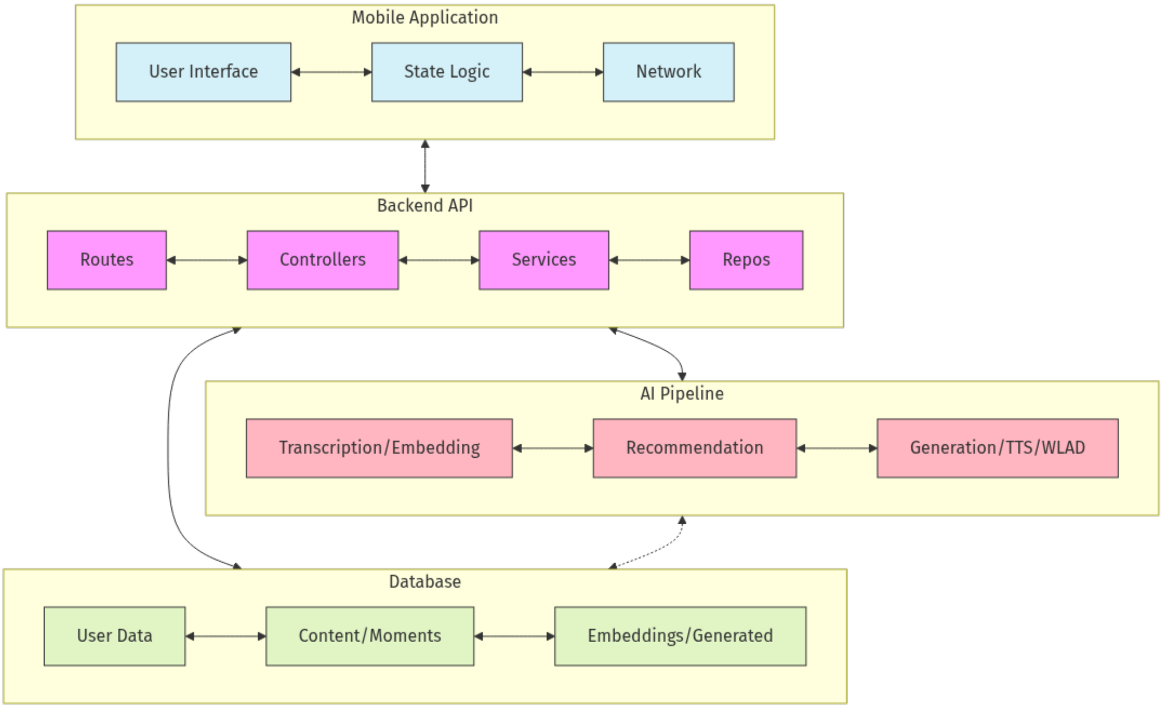

Figure 4.1: High-level architecture of the wlad system showing major components and their interactions.

4.1.2 Architectural Patterns

The architecture employs several key patterns:

- **Microservices**: AI pipeline components are implemented as specialized services.
- **Model-View-Controller**: The mobile application separates data, UI, and business logic.
- **Repository Pattern**: The backend abstracts data access to the database.
- **Pub/Sub**: Asynchronous communication handles long-running processes.
- **API Gateway**: The backend API routes and aggregates requests and responses.
- **Cross-Platform Design**: The system adapts to different device capabilities while maintaining consistent functionality.

4.2 Data Model

The data model defines the structure of information stored within the system, including user data, meditation content, reflective moments, embeddings, and generated meditations.

4.2.1 Core Entities

The primary entities are:

- **User**: System users with authentication details and roles.
- **Meditation**: Meditation sessions with metadata, content locations, and embedding vectors.
- **Moment**: User-created reflective moments with text content, metadata, and audio references.
- **Query**: User expressions forming the basis for recommendations.
- **Generated Meditation**: AI-generated meditation content based on user specifications.

The Moment entity has a particularly important role in the system, as it represents the user's personal reflective content. Each moment includes text content, category, audio reference and user association.

4.2.2 Vector Storage

The MongoDB database stores embedding vectors alongside meditation documents, supporting vector similarity operations, cosine similarity as the primary distance metric, integrated storage of vectors with document data, and metadata filtering during searches.

4.3 AI Pipeline Design

The AI pipeline provides the core intelligence of wlad through five main components: audio transcription, text embedding, similarity matching, meditation content generation, and text-to-speech conversion.

4.3.1 Audio Transcription Pipeline

The audio transcription pipeline converts meditation audio into text via:

1. **Audio Preprocessing**: Normalization and segmentation.
2. **Whisper Model Inference**: Generating text transcriptions.

3. **Post-processing**: Combining segments and verifying punctuation.

4. **Storage**: Storing the transcript with meditation metadata.

4.3.2 Text Embedding Generation

The text embedding pipeline converts transcripts and queries into vector representations:

1. **Text Preprocessing**: Cleaning and tokenizing input text.

2. **Embedding Model Inference**: Generating vectors using OpenAI's text-embedding-ada-002.

3. **Normalization**: L2-normalizing vectors for consistent similarity calculations.

4. **Vector Storage**: Storing embeddings alongside meditation documents in MongoDB.

4.3.3 Recommendation Engine

The recommendation engine matches queries with relevant content:

1. **Query Processing**: Converting user input to embedding vectors.

2. **Similarity Search**: Comparing against meditation embeddings using cosine similarity.

3. **Result Ranking**: Ranking meditations by similarity scores.

4. **Response Generation**: Returning ranked recommendations.

$$\text{similarity}(A, B) = \cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4.1}$$

4.3.4 Meditation Generation Pipeline

The meditation generation pipeline creates custom meditation content:

1. **User Input Processing**: Analyzing the user's needs and preferences.

2. **Parameter Specification**: Setting the duration attribute.

3. **Content Generation**: Creating meditation text using GPT-3.5 Turbo.

4. **Storage and Presentation**: Saving generated content and presenting it to the user.

4.3.5 Text-to-Speech Pipeline

The text-to-speech pipeline converts reflective moments to audio:

1. **Text Preparation**: Formatting and optimizing text for speech conversion.
2. **TTS Processing**: Converting text to high-quality speech using OpenAI's TTS API.
3. **Audio File Management**: Storing the generated audio with appropriate file naming.
4. **Cross-Platform URL Generation**: Creating platform-appropriate URLs for audio access.

This pipeline enables users to listen to their reflective moments, transforming personal text-based reflections into engaging audio experiences.

4.4 Reflective Moments Model

The wlad model for reflective moments represents a key innovation in this project - a specialized system designed for generating thoughtful, reflective content.

4.4.1 Model Architecture

The wlad reflective moments model uses a retrieval-based system that matches user topics to appropriate pre-written reflective content. At its core, the model uses TF-IDF vectorization to convert both the training corpus and user queries into mathematical representations. Cosine similarity calculations then identify the most relevant content based on the user's specified topic and optional category.

The model organizes content across seven distinct categories (mindfulness, gratitude, wisdom, inspiration, self-compassion, growth, and perspective), allowing for targeted content retrieval.

```python
# From train_moment_model.py
# Calculate similarity between user query and reflective moments
def generate_moment(self, topic: str, category: str = None):
    # Filter by category if provided
    candidate_moments = self.moments
    if category:
        candidate_moments = [m for m in self.moments
                             if m["category"] == category]
        if not candidate_moments:
            logger.warning(f"No moments found for category: {category}")
```

```python
            candidate_moments = self.moments

    # If we have a topic, find the most similar
    if topic:
        # Vectorize the input topic
        topic_vector = self.vectorizer.transform([topic])

        # Get topics for the filtered moments
        filtered_indices = [i for i, m in enumerate(self.moments)
                            if m in candidate_moments]

        # Extract vectors for the filtered topics
        filtered_vectors = self.topic_vectors[filtered_indices]

        # Calculate similarity
        similarities = cosine_similarity(topic_vector, filtered_vectors)
    [0]

        # Find the best match
        best_idx = np.argmax(similarities)
        chosen_moment = candidate_moments[best_idx]
    else:
        # If no topic, pick randomly from the category
        chosen_moment = random.choice(candidate_moments)

    return chosen_moment
```

Listing 4.1: Similarity-based retrieval from the wlad Reflective Moments Model

4.4.2 Training Methodology

The development of the wlad reflective moments model involved careful curation of a CSV dataset containing high-quality reflective content. Each entry includes a category, topic descriptor, and reflective text. The training process involves processing this dataset to create TF-IDF vectors for each reflection, which are then stored for later retrieval.

```python
# From train_moment_model.py - TF-IDF Training implementation
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import pickle
import os

class MomentModel:
```

```
    def __init__(self, data_path):
        self.data_path = data_path
        self.vectorizer = None
        self.topic_vectors = None
        self.moments = None

    def train(self):
        """Train the model by loading data and creating TF-IDF vectors"""
        # Load moments from CSV
        df = pd.read_csv(self.data_path)

        # Prepare data structures
        topics = df['topic'].tolist()
        categories = df['category'].tolist()
        texts = df['text'].tolist()

        # Create moments list
        self.moments = []
        for i in range(len(topics)):
            self.moments.append({
                'topic': topics[i],
                'category': categories[i],
                'text': texts[i]
            })

        # Create and fit TF-IDF vectorizer
        self.vectorizer = TfidfVectorizer(
            min_df=2,            # Minimum document frequency
            max_df=0.95,         # Maximum document frequency
            ngram_range=(1, 2),  # Consider both unigrams and bigrams
            stop_words='english' # Remove English stop words
        )

        # Fit and transform the topics to create vector representations
        self.topic_vectors = self.vectorizer.fit_transform(topics)

        print(f"Model trained on {len(topics)} moments across {len(set(
categories))} categories")
        print(f"Vocabulary size: {len(self.vectorizer.vocabulary_)}")

    def save(self, model_path):
        """Save the trained model to disk"""
        with open(model_path, 'wb') as f:
            pickle.dump({
                'vectorizer': self.vectorizer,
                'topic_vectors': self.topic_vectors,
                'moments': self.moments
```

```
        }, f)
    print(f"Model saved to {model_path}")
```

Listing 4.2: TF-IDF Vectorization for the wlad Reflective Moments Model

This approach offers several advantages, including rapid deployment, consistent quality, and easy extension by simply adding new entries to the CSV dataset.

4.4.3 Performance Characteristics

The wlad reflective moments model exhibits several performance characteristics that make it well-suited for its application:

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \cdot \text{IDF}(t, D) = f_{t,d} \cdot \log \frac{N}{|\{d \in D : t \in d\}|} \tag{4.2}$$

Where $\text{TF}(t, d)$ is the term frequency of term t in document d, and $\text{IDF}(t, D)$ is the inverse document frequency of term t in corpus D. This formula transforms text documents into numerical vectors that capture the importance of terms.

$$X \approx U \Sigma V^T \tag{4.3}$$

Singular Value Decomposition (SVD) decomposes the term-document matrix X into three matrices: U containing left singular vectors, Σ containing singular values, and V^T containing right singular vectors. This enables dimensionality reduction while preserving semantic relationships.

$$\hat{d} = \frac{d}{||d||} = \frac{d}{\sqrt{\sum_{i=1}^{n} d_i^2}} \tag{4.4}$$

Document vectors are normalized to unit length to ensure fair comparison during similarity calculations, regardless of document length.

```
# From predict.py - Prediction implementation for the reflective moments
   model
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
import random
import logging

logger = logging.getLogger(__name__)

class MomentPredictor:
    def __init__(self, model):
        self.vectorizer = model['vectorizer']
        self.topic_vectors = model['topic_vectors']
        self.moments = model['moments']
```

```
        logger.info(f"Predictor initialized with {len(self.moments)} moments")

    def predict(self, topic, category=None, top_k=5):
        """Find the most relevant moments based on topic and optional category"""
        # Filter by category if provided
        candidate_moments = self.moments
        if category:
            candidate_moments = [m for m in self.moments
                               if m["category"].lower() == category.lower()]
            if not candidate_moments:
                logger.warning(f"No moments found for category: {category}")
                candidate_moments = self.moments

        # If no topic provided, return random selection from category
        if not topic or topic.strip() == '':
            selected = random.choice(candidate_moments)
            return {
                'text': selected['text'],
                'category': selected['category'],
                'topic': selected['topic'],
                'similarity': 1.0
            }

        # Vectorize the topic query
        topic_vector = self.vectorizer.transform([topic])

        # Get indices for the filtered moments
        indices = [i for i, m in enumerate(self.moments)
                  if m in candidate_moments]

        # Calculate similarity
        similarities = []
        for idx in indices:
            sim = cosine_similarity(
                topic_vector,
                self.topic_vectors[idx]
            )[0][0]
            similarities.append((idx, sim))

        # Sort by similarity (highest first)
        similarities.sort(key=lambda x: x[1], reverse=True)

        # Return top_k results
```

```
        results = []
        for i in range(min(top_k, len(similarities))):
            idx, sim = similarities[i]
            moment = self.moments[idx].copy()
            moment['similarity'] = float(sim)
            results.append(moment)

        return results[0] if results else None
```

Listing 4.3: Similarity Calculation and Prediction for the wlad Reflective Moments Model

The performance of the system is characterized by:

- **Speed**: Near-instant retrieval of reflective content due to pre-computed TF-IDF vectors
- **Memory Efficiency**: Compact representation of the model with optimized vector storage
- **Accuracy**: Effective matching of user topics to relevant reflective content
- **Adaptability**: Category-based filtering to target specific emotional contexts

4.4.4 Integration and Deployment

The wlad reflective moments model is integrated into the application architecture through an API interface that accepts user input (topic and optional category) and returns appropriate reflective content.

The deployment process includes model training (processing the CSV dataset to create TF-IDF vectors), API implementation, response formatting, and text-to-speech integration. The system's architecture is designed for easy updates by simply extending the CSV dataset.

For meditation content generation, the system uses GPT-3.5 Turbo, providing fully generative capabilities while using the more efficient retrieval-based system for reflective moments.

4.5 User Interface Design

The user interface focuses on creating an intuitive, calming experience aligned with meditation's therapeutic nature.

4.5.1 Design Principles

Core principles include simplicity (minimizing cognitive load), calmness (using soothing visuals), accessibility (ensuring usability for people of various abilities), responsiveness (adapting to different device sizes) and progressive disclosure (presenting information in layers).

4.5.2 Key Screens and Flows

Primary user flows include recommendation flow (from expressing needs to viewing suggestions), generation flow (creating custom guided meditations and reflective moments), exploration flow (browsing content), playback flow (experiencing meditations) and administration flow (managing content).

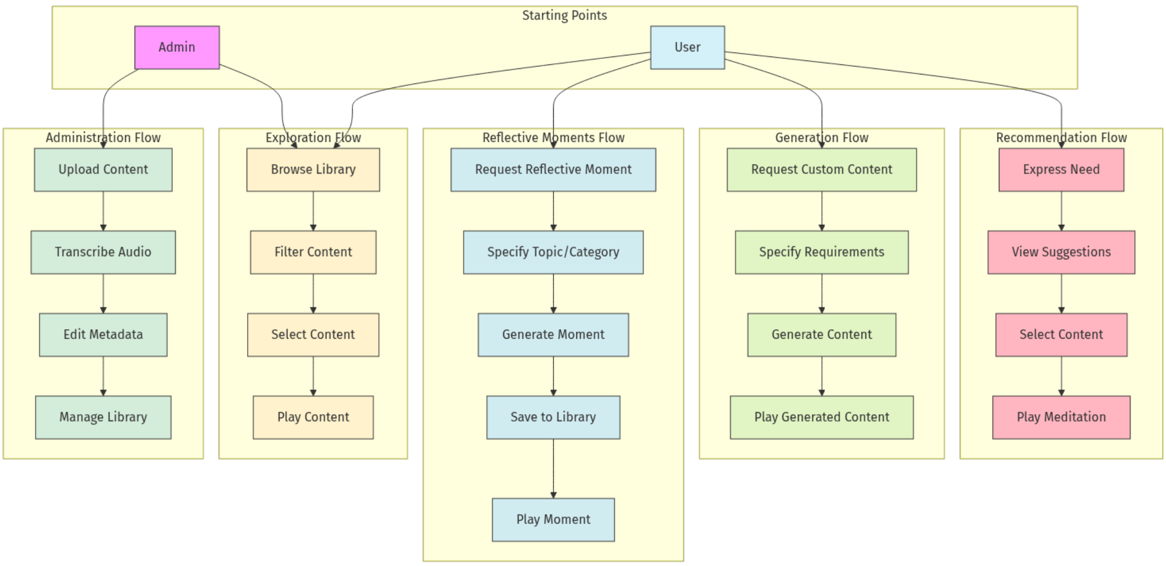

Figure 4.2: Key user flows through the wlad application.

4.6 Security and Privacy Design

Security and privacy are fundamental aspects of the system design, featuring JWT-based authentication with role-based access control, encryption at rest and in transit with data minimization, and API security including input validation, rate limiting and security headers.

4.7 Scalability Considerations

The architecture supports growth through horizontal scaling (stateless services deployed across multiple instances), database scaling (sharding and connection pooling), caching strategy (API responses and media files), and resource optimization (asynchronous processing and model pooling).

This design ensures wlad can efficiently serve a growing user base and content library while maintaining performance and reasonable operational costs.

Chapter 5

Technical Implementation

This chapter details the practical implementation of the wlad system, translating the architectural design into functional code and deployed components.

5.1 Development Environment and Tools

The development of wlad utilized a modern, integrated environment designed to support collaborative work, version control, and testing.

The primary technologies selected for implementation included:

- **Frontend**: Flutter 3.10+ for cross-platform mobile application development, with Dart 3.0+ as the programming language. The Provider package managed state, while just_audio handled audio playback.

- **Backend**: FastAPI served as the web framework for REST API development, with Python 3.9+ as the programming language. Pydantic handled data validation, while Motor provided asynchronous MongoDB access.

- **AI Components**: OpenAI API for embeddings (text-embedding-ada-002), content generation (GPT-3.5 Turbo) and text-to-speech conversion, with Whisper handling speech-to-text transcription. A custom wlad model was developed for generating reflective moments using TF-IDF and cosine similarity techniques.

- **Database**: MongoDB stored all application data including user profiles, meditation metadata, reflective moments and vector embeddings.

- **Containerization**: Docker with docker-compose orchestrated the deployment of microservices, ensuring consistent environments across development and production, with dedicated containers for the backend API, wlad model service, and database.

- **Development Tools**: Git with GitHub for source code management, following a feature branch workflow.

5.2 Frontend Implementation

The Flutter application followed a layered architecture separating presentation, business logic, data access, and utilities.

5.2.1 Application Structure and State Management

The frontend codebase was organized according to feature-first principles, with a clear separation of models, providers, screens, and services. State management utilized the Provider package, implementing a reactive programming approach where each major feature had a dedicated provider class extending ChangeNotifier.

Key providers included:

- **AuthProvider**: Managing user authentication state and credentials
- **MeditationProvider**: Handling meditation data loading and playback state
- **MomentsProvider**: Managing reflective moments creation, storage, and audio conversion
- **MeditationGeneratorProvider**: Managing the meditation generation process
- **SearchProvider**: Handling user queries and search results

5.2.2 User Interface and Experience

The UI implementation focused on creating an intuitive, calming user experience with a consistent design language implemented through Flutter's theming system. A custom theme defined a carefully selected color palette, typography and component styles that evoked tranquility and focus.

Key screens in the application included:

- **Input Screen**: Allowing users to enter their needs and view recommendations
- **Generation Screen**: Providing controls for custom meditation generation
- **Meditation Player**: Displaying meditation content with playback controls
- **Moments Screen**: Showing the user's personal reflective moments library

- **Moment Creation Screen**: Interface for creating and editing reflective moments
- **Moment Player Screen**: Specialized player for reflective moment audio
- **Meditation Library**: Showing available meditations for browsing
- **Admin Screens**: Enabling content management for administrators

5.2.3 API Integration and Audio Playback

API integration used the http package for network requests, with custom wrappers handling authentication, error management and offline support. A centralized ApiClient class managed request formatting, authentication token handling, and response parsing.

Meditation and reflective moment audio playback was implemented using the just_audio package, supporting standard playback control and progress tracking. The implementation included:

- **Navigation Controls**: Forward and rewind buttons allowed precise navigation
- **Progress Tracking**: A slider showed playback progress and allowed jumping to specific points
- **Cross-Platform Support**: Special handling for web platform audio playback using dedicated audio routes and CORS headers

The player interface displayed both audio controls and the text content of reflective moments, allowing users to read along while listening.

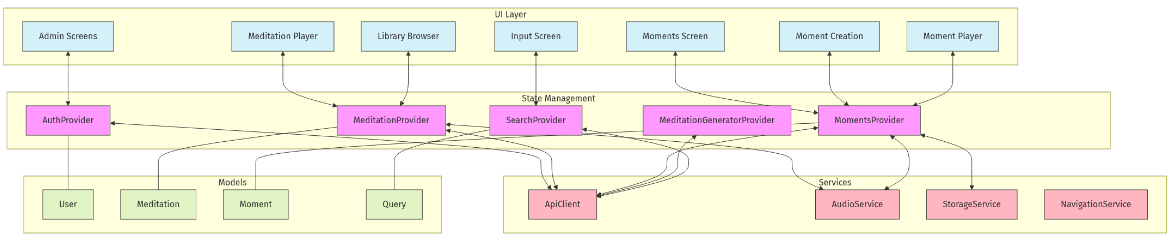

Figure 5.1: Key components of the Flutter frontend implementation showing the relationship between UI, state management, and API services.

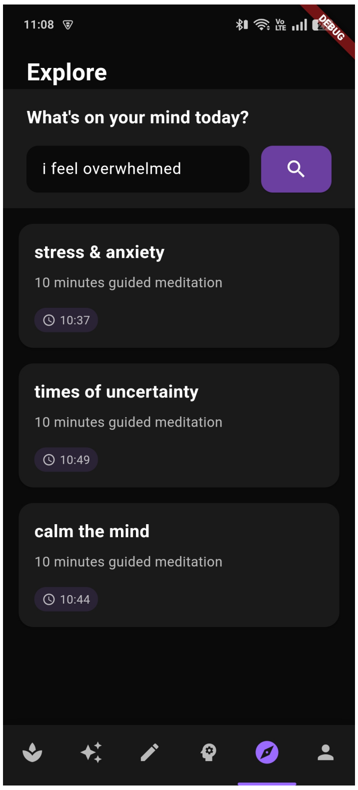

Figure 5.2: Input Screen: Users can describe their needs or feelings to get personalized recommendations

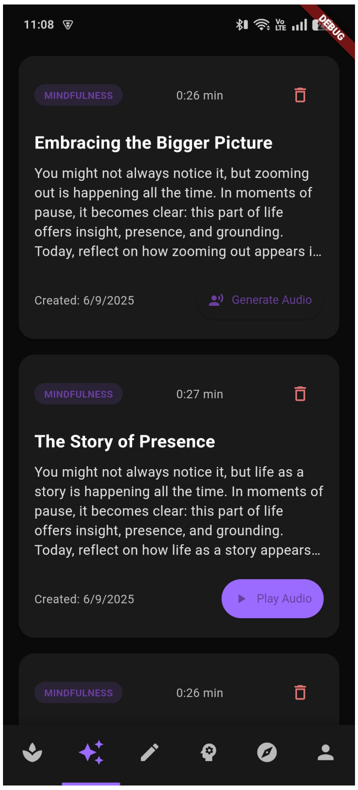

Figure 5.3: Reflective Moments Screen: Users can view and manage their personal reflective content

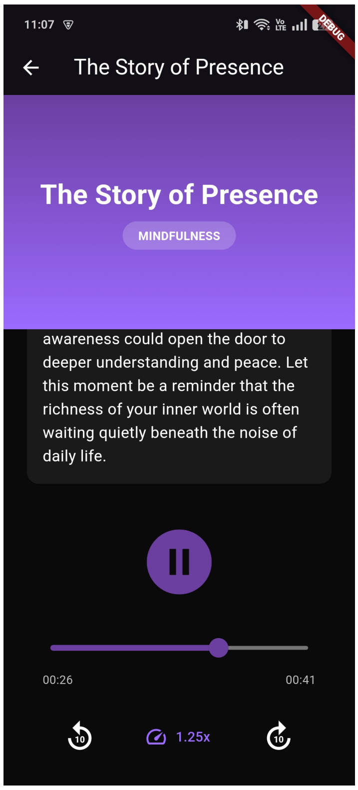

Figure 5.4: Audio Player: Unified player interface for both meditation content and reflective moments

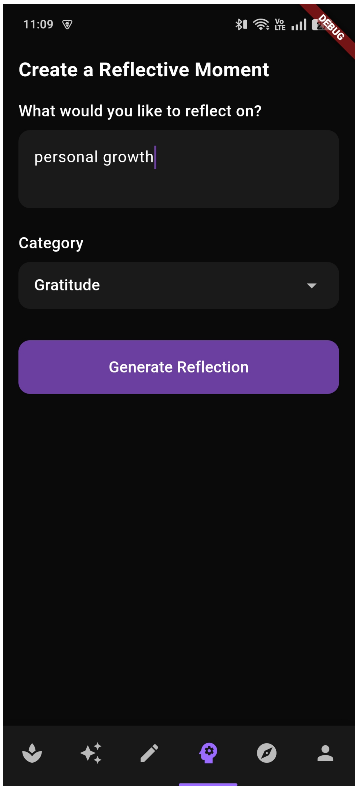

Figure 5.5: Moment Creation: Interface for users to write and categorize their reflective moments

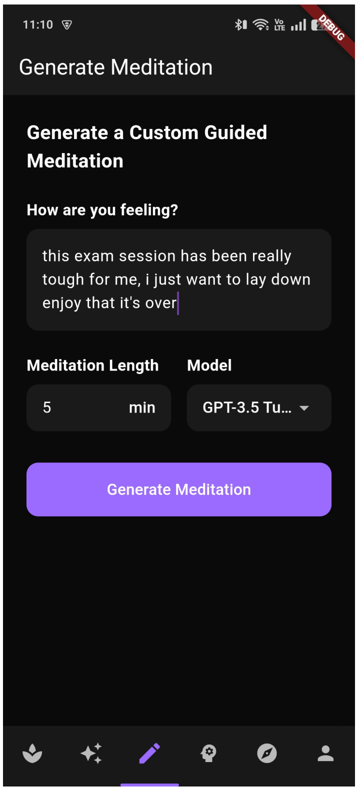

Figure 5.6: Generate Meditation: Interface for users to create custom meditation content

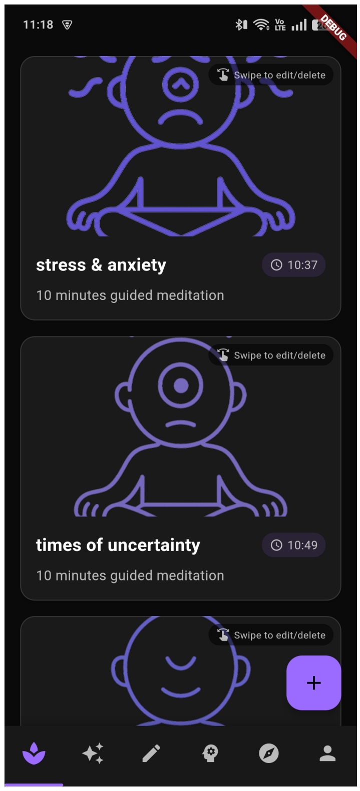

Figure 5.7: Meditations Screen: Interface for displaying and managing saved content

5.3 Backend Implementation

The backend implementation provided the API layer, business logic, and data access for the wlad system. Built with FastAPI, the application structure followed a modular approach with separate components for routes, models, services, and authentication.

5.3.1 Application Structure and API Endpoints

The backend codebase was organized following a domain-driven structure. API endpoints were organized as FastAPI routers by feature area, with Pydantic models validating all request inputs and standardizing response formats.

Endpoint	Method	Description
/auth	POST	User registration, login, and token management
/meditations	GET, POST	CRUD operations for meditation content
/moments	GET, POST	CRUD operations for personal reflective moments
/moments/:id	GET, PUT, DELETE	Get, update, or delete a specific reflective moment by ID
/moments/:id/convert-to-audio	POST	Convert a text moment to speech
/audio	GET	Serving audio files with proper CORS headers for web playback
/search	POST	Processing user queries and returning meditation recommendations
/generate	POST	Creating custom meditation content
/api/health	GET	Health check for the wlad model server
/api/categories	GET	Retrieve available reflective moment categories
/api/generate	POST	Generate a reflective moment based on topic and category

Table 5.1: API Endpoints in the wlad System

5.3.2 Authentication and Data Access

The authentication system implemented JWT-based token authentication with bcrypt handling password hashing and role-based authorization controlling access to administrative endpoints.

Database access was handled through asynchronous operations using Motor for MongoDB, ensuring non-blocking I/O for optimal performance. The application used connection pooling to efficiently manage database connections.

5.3.3 Media File Handling

Media file handling supported multiple types of audio content:

- **Meditation Audio**: Administrators could upload meditation audio files through multipart form uploads
- **Generated Audio**: Text-to-speech services created audio files from reflective moment text
- **Audio Serving**: A dedicated audio route with proper CORS headers served audio files to both mobile and web clients

The system implemented a structured storage organization with separate directories for different types of audio content. For web platform support, special attention was given to CORS headers and content types to ensure cross-browser compatibility.

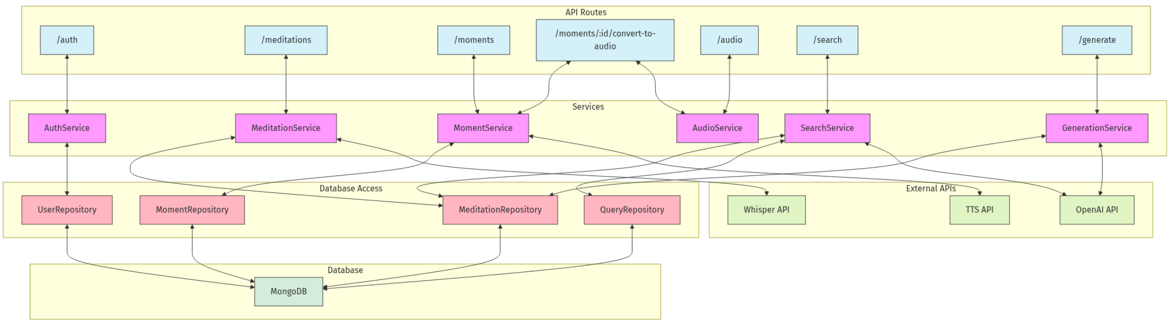

Figure 5.8: Backend architecture showing API routes, services, and database interactions.

5.4 AI Model Integration

The AI capabilities of wlad were implemented through integration with external APIs and a custom reflective moments model.

5.4.1 Transcription and Embedding Pipeline

The transcription service utilized OpenAI's Whisper model to process meditation audio files. The transcription process converted spoken meditation guidance into textual form, enabling semantic analysis and embedding generation.

For embedding generation, the system integrated with OpenAI's text-embedding-ada-002 model to create vector representations of both meditation transcripts and user queries. These embeddings were normalized and stored directly in MongoDB alongside other meditation metadata.

The recommendation engine matched user queries with relevant meditation content by calculating cosine similarity between embedding vectors. Results were ranked by similarity score and returned to the user.

5.4.2 Text-to-Speech Conversion for Reflective Moments

The text-to-speech service converted user-created reflective moment text into audio files using OpenAI's TTS API with appropriate pacing and intonation for reflective content.

The implementation included text preprocessing, voice selection, error handling and file management. The conversion process ran asynchronously to avoid blocking the user interface with status updates to keep users informed of progress.

5.4.3 Reflective Moments Model Implementation

The reflective moments model used a simple yet effective TF-IDF and cosine similarity approach to match user queries with appropriate pre-written reflective content. The implementation included:

- **CSV Dataset**: A curated collection of high-quality reflective content organized by category and topic
- **TF-IDF Vectorization**: Converting both the training corpus and user queries to mathematical representations
- **Similarity Matching**: Using cosine similarity to identify the most relevant content based on the user's topic and optional category
- **Categorical Organization**: Content organized across seven categories (mindfulness, gratitude, wisdom, inspiration, self-compassion, growth and perspective)

The TF-IDF implementation used the scikit-learn library with the following parameter configuration:

$$\text{TF-IDF}_{config} = \begin{cases} \text{min\_df} = 2 \\ \text{max\_df} = 0.95 \\ \text{ngram\_range} = (1, 2) \\ \text{stop\_words} = \text{english} \end{cases} \tag{5.1}$$

For similarity calculations, the system implemented the cosine similarity formula using scikit-learn's optimized functions:

$$\text{score}(q, d_i) = \frac{q \cdot d_i}{||q|| \cdot ||d_i||} \quad \text{for} \quad i \in \{1, 2, \ldots, n\} \tag{5.2}$$

Where q represents the user's query vector and d_i represents the document vector for the i-th reflective moment in the dataset.

For custom meditation generation, the system used GPT-3.5 Turbo to create content based on user-specified needs with appropriate prompt engineering to guide the generation process.

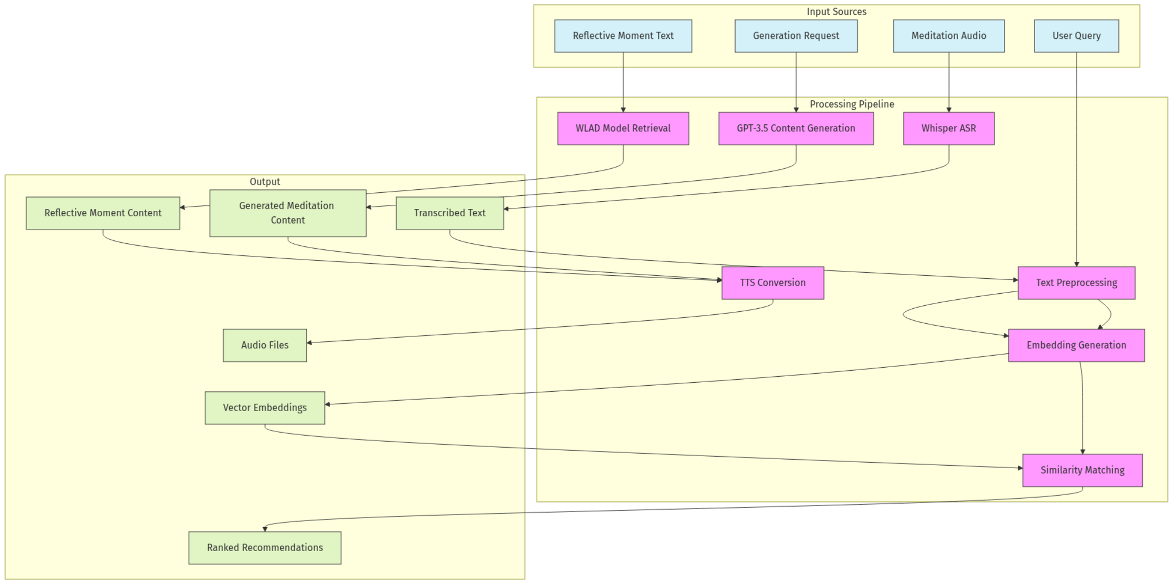

Figure 5.9: AI pipeline showing the flows for transcription, embedding generation, recommendation, content generation, and text-to-speech conversion.

5.5 Database Implementation

The database layer utilized MongoDB for both document storage and vector embeddings. MongoDB stored all structured data including user profiles, meditation metadata, user queries, generated meditations, reflective moments, and embedding vectors.

5.5.1 Collection Structure

The document schema included carefully designed collections:

- **Users**: User accounts with authentication details and role information
- **Meditations**: Pre-recorded meditation content with metadata and transcriptions

- **GeneratedMeditations**: AI-generated meditation content with associated parameters
- **Moments**: User-created reflective moments with text content, metadata and audio file references
- **Queries**: User search inputs and associated context

5.5.2 Vector Storage

For vector storage, the system stored embedding vectors directly within meditation documents as arrays of floating-point numbers, providing simplified architecture, data locality, transactional integrity and query efficiency.

5.5.3 Cross-Platform Audio Storage

The system implemented a flexible audio storage strategy to accommodate both mobile and web platforms with native file system access for mobile platforms and a dedicated audio route with CORS headers for web platforms.

Chapter 6

Future Evaluation Framework

This chapter briefly outlines potential evaluation approaches that could be applied to the wlad application in future work.

6.1 Potential Evaluation Approaches

While formal testing and evaluation were outside the scope of this thesis project, several methodologies could be valuable for future assessment of the wlad system:

- **Technical Performance Metrics**: Measuring recommendation accuracy, response times and audio processing quality.
- **User Experience Evaluation**: Assessing satisfaction, perceived relevance, usability and engagement through user studies.
- **Audio Quality Assessment**: Evaluating text-to-speech conversion quality and playback functionality.

A comprehensive evaluation would ideally combine quantitative performance metrics with qualitative user research to gain insights into both technical performance and real-world usage patterns.

6.2 Recommendation System Evaluation

The effectiveness of wlad's recommendation system could be evaluated by comparing embedding-based matching against traditional keyword and category-based approaches. Metrics such as precision, recall and mean reciprocal rank would help quantify the system's ability to deliver relevant content based on user queries.

For a rigorous assessment, a test dataset of diverse user queries with pre-labeled relevant meditations would need to be created. This would enable objective measurement of how accurately the system matches user inputs with appropriate meditation content.

6.3 Reflective Moments Evaluation

The reflective moments feature could be evaluated across several dimensions:

- **Text-to-Speech Quality**: Assessing the naturalness and clarity of generated audio
- **Content Relevance**: Measuring how well the TF-IDF and cosine similarity approach identifies appropriate reflective content
- **User Satisfaction**: Gathering feedback on the perceived value of the feature

6.4 Future Directions for Evaluation

Future work could include structured user studies with participants from diverse backgrounds to evaluate the real-world effectiveness of the wlad application. Such studies would provide valuable insights into how different user groups interact with the application and which features they find most beneficial.

Potential areas of focus for future evaluation include:

- Cross-platform performance and compatibility
- Long-term engagement and retention patterns
- Comparative analysis against other wellness applications
- Impact assessment on user wellbeing through longitudinal studies

By incorporating such evaluation methodologies in future iterations, the wlad application could be continuously refined to better meet user needs and improve overall effectiveness.

Chapter 7

Conclusions and Future Work

This chapter summarizes the research contributions, discusses the implications of the findings and outlines directions for future work.

7.1 Summary of Contributions

This thesis has presented wlad, an AI-driven mobile and web application that enhances wellness through both meditation content and personal reflective moments. The main contributions include:

- A novel approach to meditation content discovery using embedded representations of both user queries and meditation transcriptions
- An integrated system for personal reflective moments with direct text-to-speech conversion and advanced audio playback
- A cross-platform application architecture with responsive interfaces for both mobile and web experiences
- A specialized meditation and reflection generation system based on the wlad model
- A complete audio processing pipeline that converts text to high-quality speech with appropriate pacing for reflective content
- User-centric design that allows for personalized management of reflective content

7.2 Practical Implications

The findings of this research have several practical implications:

- **For Users**: The system enables both discovery of existing meditation content and creation of personal reflective moments and guided meditations with audio capabilities, potentially improving user engagement and the effectiveness of wellness practices.
- **For Content Creators**: The framework provides new opportunities for wellness content creators to make their content more accessible through audio conversion and playback features.
- **For Mental Health Applications**: The approach demonstrates how AI can be applied to mental health contexts in ways that complement traditional content delivery methods, offering more personalized and interactive experiences.
- **For Audio Content Consumption**: The variable speed playback and navigation controls show how digital wellness content can be consumed in ways that adapt to individual preferences and needs.

7.3 Limitations

Despite the promising results, several limitations should be acknowledged:

- **Text-to-Speech Quality**: While current TTS technology provides high-quality audio, it still lacks some of the nuanced emotional expression of human narrators, particularly for meditative content.
- **Web Audio Playback**: Cross-browser compatibility and CORS configurations present challenges for consistent audio playback experiences across all platforms.
- **Content Diversity**: The effectiveness of recommendations depends on having a diverse library of meditation content to match various user needs.
- **Computational Requirements**: Generation of both text content and audio requires significant computational resources, which could impact scalability.
- **Mobile Platform Limitations**: Native feature support varies across platforms, requiring adaptations for optimal user experience on each device type.

7.4 Future Work

The development of wlad opens several promising avenues for future research and development that could extend and enhance the system's capabilities. Refining the wlad generation model represents a primary direction for future work. This refinement would focus on improving the quality and diversity of reflective moments, potentially incorporating more personalized elements based on user history and preferences.

Another compelling direction involves enhancing the text-to-speech capabilities to produce more naturalistic meditation audio. This could include more sophisticated voice models with better emotional expression, more natural pacing, and the ability to incorporate appropriate background sounds that complement the reflective content. Such enhancements would further bridge the gap between generated content and professional recordings.

The audio playback experience could be extended with additional features such as:

- **Visualization elements** that synchronize with the audio to provide visual cues for breathing or focus
- **Customizable background sounds** that users can mix with their reflective moment audio
- **Progress tracking** that monitors usage patterns and helps users establish consistent wellness routines
- **Social sharing options** that allow users to share their reflective moments with trusted connections

The recommendation system could be enhanced to incorporate feedback from moment interactions, building a more personalized understanding of each user's preferences and needs. This could lead to increasingly relevant suggestions for both pre-recorded meditations and generated reflective content.

Cross-cultural adaptation would expand the system's reach by supporting multiple languages for both the interface and content generation. This would require not just translation but cultural adaptation of the underlying models to generate appropriate reflective content for diverse linguistic and cultural contexts.

Finally, conducting effectiveness studies would provide valuable insights into how regular use of personalized reflective moments impacts various wellness outcomes. These studies could assess factors such as stress reduction, emotional regulation, sleep quality and overall well-being, providing empirical validation for the approach and guiding future refinements.

7.5 Concluding Remarks

The wlad project demonstrates how AI technologies can be applied to create more personalized, accessible mental well-being tools. By combining content discovery with personal reflective moments and advanced audio capabilities, the system offers users unprecedented flexibility in managing their wellness journey.

The implementation of direct audio generation and playback for reflective moments represents a significant advancement over traditional category-based wellness apps. This feature transforms personal reflections from static text to engaging audio experiences that can be consumed at the user's preferred pace and style.

As digital wellness practices continue to evolve, there are rich opportunities for innovation at the intersection of artificial intelligence, audio technology and mental health. This work represents a step toward more intuitive, contextually aware digital well-being tools that better serve diverse human needs and preferences. By putting users in control of both content discovery and creation, wlad empowers individuals to take a more active role in their mental wellness journey.

Bibliography

[AT05] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[BCMK19] Ruth Baer, Catherine Crane, Edward Miller, and Willem Kuyken. Potential for harm in mindfulness-based interventions: A reconsideration. *American Psychologist*, 74(7):878–885, 2019.

[BHA+21] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[CZZO22] Matthew Cohn, Cheng Zhuang, Jing Zhu, and Sageev Oore. Prosodyspeech: Controllable expressive text-to-speech for mental health applications. In *Proceedings of the Annual Conference of the International Speech Communication Association*, pages 3803–3807, 2022.

[DSCP20] Yashar Deldjoo, Markus Schedl, Paolo Cremonesi, and Gabriella Pasi. Recommender systems leveraging multimedia content. *ACM Computing Surveys*, 53(5):1–38, 2020.

[ERN+18] Marcos Economides, Kristian Ranta, Anna Nazander, Outi Hilgert, Philippe R Goldin, Anu Raevuori, and Valerie Forman-Hoffman. Improving sleep quality using guided meditation with calm: A randomized-controlled trial. *Journal of Medical Internet Research*, 20(11):e12332, 2018.

[GDC23] Jianfeng Gao, Zhuyun Dai, and Weizhu Chen. Embedding-based retrieval for natural language queries. *Transactions on Machine Learning Research*, 2023.

[Gra23] Grand View Research. Meditation app market size, share & trends analysis report by type, by service type, by region, and segment forecasts, 2023-2030. Technical report, Grand View Research, 2023.

[Gro15] James J Gross. Emotion regulation: Current status and future prospects. *Psychological Inquiry*, 26(1):1–26, 2015.

[GRSD22] Simon B Goldberg, Kevin M Riordan, Shufang Sun, and Richard J Davidson. The empirical status of mindfulness-based interventions: A systematic review of 44 meta-analyses of randomized controlled trials. *Perspectives on psychological science*, 17(1):108–130, 2022.

[HHE+17] Lara Hilton, Susanne Hempel, Brett A Ewing, Eric Apaydin, Lea Xenakis, Sydne Newberry, Ben Colaiaco, Alicia Ruelaz Maher, Roberta M Shanman, Melony E Sorbero, et al. Mindfulness meditation for chronic pain: systematic review and meta-analysis. *Annals of Behavioral Medicine*, 51(2):199–213, 2017.

[HPES20] Jennifer Huberty, Megan Puzia, Ryan Eckert, and Michael Sullivan. Cancer patients' and survivors' experiences using a meditation mobile app for anxiety reduction. *JMIR Cancer*, 6(2):e19154, 2020.

[HSG+22] Jennifer Huberty, Michael Sullivan, Jeni Green, Jonathan Kurka, Jenn Leiferman, Katherine Gold, and Joanne Cacciatore. Testing a workplace mindfulness meditation app: Randomized controlled trial. *JMIR mHealth and uHealth*, 10(4):e30272, 2022.

[KNSP15] Jonathan W Kanen, Raheela Nazir, Karim Sedky, and Basant K Pradhan. Effects of mindfulness-based interventions on sleep quality: A systematic review. *Evidence-Based Complementary and Alternative Medicine*, 2015:902708, 2015.

[KTP+19] Kira Kretzschmar, Helene Tyroll, Gabriela Pavarini, Arianna Manzini, and Ilina Singh. Can your phone be your therapist? young people's ethical perspectives on the use of fully automated conversational agents (chatbots) in mental health support. *Biomedical Informatics Insights*, 11:1178222619829083, 2019.

[KYS+23] Ye Lim Kim, Jaeyoon Yang, Yunseo Sung, Seungyeon Lee, Doheum Lee, and Bongwon Suh. Enhancing mobile meditation app user experience: Exploring needs, barriers, and design implications. *International Journal of Human-Computer Studies*, 171:102971, 2023.

[LC18] Emily K Lindsay and J David Creswell. Mindfulness and emotion regulation: Insights from neurobiological, psychological, and clinical studies. *Nature Reviews Neuroscience*, 19(7):458–466, 2018.

[LD97] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

[MKHS15] Madhavan Mani, David J Kavanagh, Leanne Hides, and Stoyan R Stoyanov. Review and evaluation of mindfulness-based iphone apps. *JMIR mHealth and uHealth*, 3(3):e4328, 2015.

[MMLH+21] Nicole Martinez-Martin, Owen Lewis, Desiree Hellman, Lisa Minor, Mildred Billingsley, Matthew P Lungren, and Brent Mittelstadt. Ethics of artificial intelligence in psychiatry: An ethical framework for evaluating automated and autonomous systems. *European Psychiatry*, 64(1):e47, 2021.

[MNL+21] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Semantics-aware recommender systems exploiting linked open data and graph-based features. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 174–183. Springer, 2021.

[Nor13] Don Norman. *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books, New York, 2013.

[PB07] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. *The Adaptive Web*, pages 325–341, 2007.

[PPC25] Amalia Maria Postolache, Ioan Daniel Pop, and Adriana Mihaela Coroiu. Insightful mental health tool for students. *Proceedings of the International Conference on Health Informatics*, pages 133–140, 2025.

[RG19] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992, 2019.

[RGF+23] Ariel Rao, Sharath Chandra Guntuku, Andrew Fiore, Sang Yup Jeon, Richa Sinha, and David A Asch. The state of mental health applications in us adults. *JAMA Psychiatry*, 80(9):942–944, 2023.

[RKX+23] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. *International Conference on Machine Learning*, pages 28492–28518, 2023.

[SBM+20] Emma M Seppälä, Christina Bradley, Julia Moeller, Leilah Harouni, Dhruv Nandamudi, and Marc A Brackett. Promoting mental health and psychological thriving in university students: a randomized controlled trial of three well-being interventions. *Frontiers in psychiatry*, 11:534776, 2020.

[SFHS07] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. *The Adaptive Web*, pages 291–324, 2007.

[TFH+18] John Torous, Joseph Firth, Kit Huckvale, Mark Erik Larsen, Theodore D Cosco, Rosemary Carney, Simon Chan, Abhishek Pratap, Peter Yellowlees, Til Wykes, et al. Towards a framework for evaluating mobile mental health apps. *World Psychiatry*, 17(2):236–236, 2018.

[TQSL22] Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. Naturalspeech: End-to-end text to speech synthesis with human-level quality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9271–9287, 2022.

[VHT21] Arun Vaidyam, John Halamka, and John Torous. Artificial intelligence in mental health: Current applications and future directions. *JMIR Mental Health*, 8(10):e30091, 2021.

[Wor22] World Health Organization. Mental health and covid-19: Early evidence of the pandemic's impact. *Scientific Brief*, 2022.

[WSZ+22] Zhizheng Wang, Sumeet Singhal, Tianyi Zhu, Yao Xu, and Simon King. Towards robust speech recognition for mindfulness applications. *IEEE Journal of Biomedical and Health Informatics*, 26(10):4918–4929, 2022.

[YDMS23] Mengxue Yin, Alexandra M DiGiorgio, David C Mohr, and Stephen M Schueller. Language model embeddings for therapeutic content matching in digital mental health interventions. *Journal of Medical Internet Research*, 25(8):e46371, 2023.

[ZLW+22] Lei Zhang, Bo Liu, Jun Wang, Shuiguang Deng, and Ke Chen. Emotional context-aware recommender systems. *Information Sciences*, 604:161–183, 2022.