AstroMood Project Design Document

Sümeyye Sıla Altay 231101077
Esmanur Ulu 231101024
Zeynep Yetkin 231101042
Hazal Epözdemir 231101037

Document-Specific Task Matrix:

Task	**Team Member Responsible**
System Overview	Zeynep Yetkin
Implementation Details	Hazal Epözdemir
Use Case Support in Design	Esmanur Ulu
Design Decisions	Sümeyye Sıla Altay

1. System Overview

1.1 Project Description

AstroMood is an AI-powered website that analyzes users' zodiac signs and daily moods to provide personalized recommendations. By combining astrology and artificial intelligence, the platform aims to enhance users' daily motivation and well-being. The system evaluates users' emotional states in real time and offers tailored advice on personal development, stress management, and goal setting.

The project includes key features such as horoscope retrieval, mood analysis, AI-driven recommendations (including book, movie, and music suggestions), and an intuitive user interface. It integrates with platforms like Spotify to deliver personalized playlists.

The development process involves machine learning techniques, user feedback mechanisms, and continuous improvements to ensure accuracy and engagement. AstroMood's success will be measured by user satisfaction, adoption rates, and the effectiveness of its personalized recommendations.

1.2 System Architecture

General Structure: The project will be developed using a layered architecture, consisting of two main layers:

- **Presentation Layer (Frontend)**
- **Business Logic Layer (Backend)**

Main Components:

- **User Interface:** A dynamic and responsive interface built with React.js.
- **API Server:** RESTful APIs built using Node.js and Express.
- **Communication Protocols:** API requests will be in JSON format, and WebSockets may be used for real-time updates.

1.3 Technology Stack

- **Frontend:** React.js, HTML, CSS, JavaScript
- **Backend:** Python
- **Others:** GitHub (Version Control)

2. Implementation Details

2.1 Code Structure

File and Folder Organization: The project is structured under the `src/` directory with the following subdivisions:

- `components/` – UI components
- `services/` – API and business logic
- `assets/` – Visuals and static files
- `tests/` – Unit and integration tests
- `PA1Docs/` – Project Definition, Project Plan, Requirements
- `PA2Docs/` – Project Design Document, Quality Assurance (QA) Plan

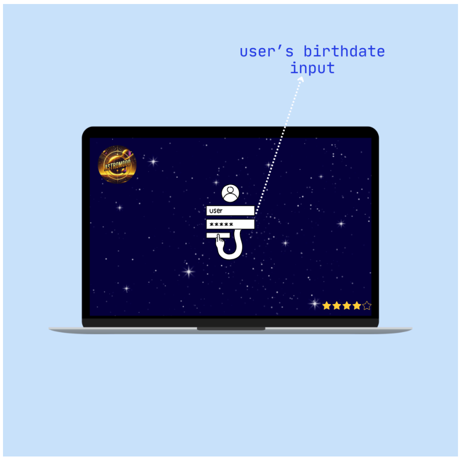
Modular Structure: Each module is developed independently and integrated as needed.

2.2 Main Components & Business Logic

- **Astrological Calculations:** Determining the user's zodiac sign based on their birth date.
- **Mood Analysis:** Analyzing users' moods based on their input.
- **Recommendation Engine:** Providing personalized recommendations based on astrological data and mood analysis.
- **API Design:** RESTful endpoints, such as:
  - `POST /api/astrology`
  - `GET /api/recommendations`

2.3 Visual Interfaces

Wireframes and Mockups: The updated versions of the PA1 screen designs should be added.



Main Pages:

- **Homepage:** Displays user login and feedback corner with stars.
- **Mood Input Page:** Allows users to enter their current mood.
- **Recommendation Page:** Displays personalized recommendations based on mood and astrological data.

---

3. Use Case Support

3.1 Selected Use Cases

The project focuses on four key use cases:

1. **User Birth Date Input & Zodiac Calculation:**
   - Users can enter their birth date.
   - The system calculates their zodiac sign based on the provided birth date.
2. **Mood Input:**
   - The system must allow users to input their current mood daily without storing historical data.
3. **Personalized Recommendations:**
   - The system generates personalized suggestions based on the user's zodiac sign and mood.
4. **Quote & Motivation Message Generator:**
   - The system displays motivational quotes based on the user's mood.

3.2 Requirement Mapping

Each use case is mapped to specific functional requirements:

Use Case - Functional Requirements Matching:

1. User Birth Date Input & Zodiac Calculation: Requirement 1 & 5
2. Mood Input: Requirement 2
3. Personalized Recommendations: Requirement 2, 3 & 4
4. Quote & Motivation Message Generator: Requirement 8 & 3

3.3 Use Case Design

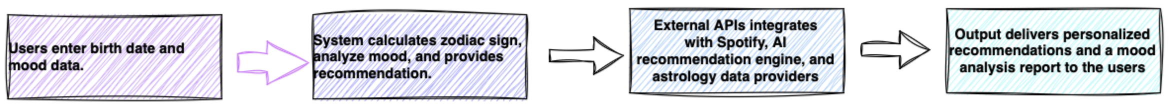
Data Flow Explanation (Initial Version):
The system consists of three layers:

1. **Frontend (React.js):** Handles user input and displays results.
2. **Backend (FastAPI/Flask):** Processes input, retrieves data, and applies AI models.
3. **External Services & AI Models:** Fetches horoscope data, analyzes mood, and generates recommendations.

Step-by-Step Data Flow:

1. **User Inputs Data (Frontend):**
    - The user enters their birth date and mood description.
    - The frontend sends a request to the backend.
2. **Backend Processes the Request (FastAPI/Flask):**
    - The API processes user input and triggers multiple actions:
        - Calculates the zodiac sign.
        - Calls the Horoscope API to fetch daily horoscopes.
        - Analyzes mood sentiment.
        - Calls the Recommendation System to fetch books, movies, and music.
        - Connects to the Spotify API for personalized song recommendations.
3. **Horoscope Retrieval (External API or Web Scraping):**
    - The backend either calls an astrology API or scrapes horoscope websites for predictions.
4. **AI-Driven Recommendation Engine:**
    - The system recommends books, movies, and music based on:
        - The user's zodiac sign (e.g., Leos prefer inspirational books).
        - The mood classification (e.g., Sad → Uplifting books).
5. **Quote & Motivation Message Recommendation:**
    - Two approaches are considered:
        - Using an external API for dynamic motivational quotes.
        - Displaying predefined quotes stored within the app for better control and offline availability.
    - The final decision depends on usability, technical feasibility, and integration complexity.

6. **Backend Sends Data to Frontend:**
    - The backend compiles the results and returns them in a structured JSON format.
7. **Frontend Displays Results:**
    - The React frontend updates the UI with:
        - Daily horoscope
        - AI-analyzed mood insights
        - Personalized book, movie, and music recommendations


Users enter birth date and mood data.
System calculates zodiac sign, analyze mood, and provides recommendation.
External APIs integrates with Spotify, AI recommendation engine, and astrology data providers
Output delivers personalized recommendations and a mood analysis report to the users

4. Design Decisions

4.1 Technology Comparisons

- **Frontend Framework:** React.js and Angular were compared. React.js was preferred due to its flexibility and wide ecosystem.
- **Backend Framework:** Django/FastAPI and Node.js were evaluated. Django/FastAPI was chosen for its suitability in AI and machine learning integration due to its simpler structure.

4.2 Decision Rationale

- **React.js Choice:** Chosen due to its component-based structure, extensive community support, and previous experience from PA1.
- **Django/FastAPI Choice:** Selected for its asynchronous structure, fast API development, and compatibility with the existing JavaScript ecosystem.