Overview

To successfully complete your assignment on System Requirements Specifications (SRS) and create a high-level UML-based system overview, follow these structured steps:

Prepare

1. Understanding the Project Description

- **Read and Analyze:** Carefully read the project description to understand its scope, objectives, and key features.
- **Identify Key Elements:** Highlight key elements like target users, primary functionalities, and specific constraints or goals mentioned in the project description.
- **Clarify Goals:** Ensure every team member knows the end goals of the project.

2. Gather Your Resources (Define Clearly)

- **Research Best Practices:** Look into best practices for SRS documentation and UML diagrams relevant to your project type.
- **Review Past Lectures:** Go through your lecture notes and materials to understand the expectations regarding use cases and activity diagrams.
- **Consult Templates and Examples**: Review the provided SRS template and find examples of similar projects to get an idea of standard practices and formatting.

3. Team Structure and Roles

- **Project Coordinator - Member 1**
- **Requirements Analyst - Member 2**
- **UML and System Design Specialist - Member 3**
- **SRS Document Writer - Member 4**
- **Quality Assurance and Reviewer - Member 5**

Start

4. Listing Functional Requirements

- **Gather Requirements:** List all the functional and non-functional requirements.

- **Categorize and Prioritize:** Group similar functionalities and prioritize them based on the project goals. Determine which requirements are essential and which are optional.
- **SRS Template:** Use the provided SRS template to organize these requirements.

5. Defining High-Level Objects and Developing UML Diagrams

- **Identify Key Objects**: Determine the main objects or classes in your system (e.g., User, Product, Order in an e-commerce system).
- **Develop UML Diagrams**: Developing UML (Unified Modeling Language) diagrams is a crucial step in visualizing and designing the system architecture. UML diagrams help in understanding the flow, interactions, and structure of your system.

6. Regular Team Meetings and Minutes

- **Schedule Regular Meetings**: Set up regular meetings with your team to discuss progress, challenges, and next steps.
- **Record Minutes:** Document key points, decisions, and individual responsibilities discussed in each meeting. This keeps track of progress and holds team members accountable.
- **Peer Review Sessions:** Organized by the Quality Assurance and Reviewer for cross-checking work.
- **Collaboration Tools:** Use tools like Trello, Asana, or Jira for task management and Slack or Microsoft Teams for communication

7. Finalizing and Review

- **Review and Edit:** Thoroughly review the SRS document and UML diagrams for completeness, accuracy, and clarity.
- **Peer Review:** Have team members review each other's work for additional perspectives and catch potential errors.
- **Finalize Documentation:** Ensure that the final version of the SRS document and UML diagrams are well-organized, formatted according to the template, and meet the project's requirements.

8. Presentation and Submission

- **Prepare Presentation:** If required, prepare a presentation summarizing key aspects of your SRS and UML diagrams.

- **Practice and Feedback:** Practice the presentation with your team and seek feedback to refine it.
- **Submit Assignment:** Ensure that the final documents and any required presentation materials are submitted before the deadline.

End

Tips for Success

- **Effective Communication:** Maintain clear and open communication within the team.
- **Time Management:** Allocate sufficient time for each section of the project.
- **Adaptability:** Be prepared to make changes as new information or feedback is received.

Guideline for developing SRS and creating a high-level UML-based system

Project: Music Emoji App
Project Description: Music can change the mood of the listener however there are some days that you feel low or happy but don't know what music is suitable for your mood. This project will help you choose music based on your mood. This project aims to create a mobile app that will suggest music to play based on the user's mood. Using face recognition, the system will analyze the image and suggest music according to the mood shown in the image. This will also analyze the trend of the moods to give suggestions or tips on improving good mood and health.

Prepare

1. Understanding the Project Description

1.1. Read and Analyze

- **Project Scope:** The scope of the Music Emoji App project encompasses the development of a mobile application that uses facial recognition technology to analyze a user's mood and suggest music accordingly.
    - **Development of a Facial Recognition System:** Implementing an AI-based facial recognition system capable of accurately identifying a user's emotional state.
    - **Integration with Music Libraries:** Establishing a connection with music streaming services or building an in-app music library to provide music recommendations.
    - **Mood Analysis Algorithm:** Creating an algorithm that interprets facial recognition data to assess the user's mood.
    - **Mood Trend Analysis:** Analyzing mood data over time to track trends and provide insights into the user's emotional well-being.
    - **User Interface Design:** Designing an intuitive and user-friendly interface allows users to easily use the app, view their mood trend data, and listen to suggested music.
    - **Privacy and Data Security:** Ensuring the privacy and security of user data, particularly sensitive facial recognition data and mood analysis results.
- **Project Objectives**

    - **Enhance User Mood:** To provide users with music recommendations that positively influence their mood and emotional state.
    - **Accurate Mood Detection:** To achieve high accuracy in mood detection through advanced facial recognition technology.
    - **Personalized Experience:** To offer a personalized user experience by tailoring music suggestions based on individual mood trends and preferences.
    - **User Engagement and Retention:** To engage users by offering insightful mood trend analyses and health tips, thereby increasing app usage and retention.
    - **Data Privacy and Security:** To prioritize user privacy and security, especially in handling and storing facial recognition data and personal preferences.
    - **Compatibility and Performance:** Ensuring the app is compatible with various devices and operating systems, and optimizing for high performance and reliability.
- **Key Features**
    - **Facial Recognition for Mood Detection:** The app uses facial recognition technology to analyze the user's facial expressions and determine their mood.
    - **Music Recommendation Engine:** Based on the detected mood, the app suggests a selection of music that aligns with the user's current emotional state.
    - **Mood Trend Tracking:** The app tracks the user's mood over time, providing insights and trends in their emotional well-being.
    - **Customizable User Preferences:** Users can customize their music preferences, allowing the app to provide more tailored suggestions over time.
    - **Interactive User Interface:** An easy-to-navigate interface that displays mood analysis, music suggestions, and mood trends in an engaging and accessible format.
    - **Privacy Settings and Data Management:** Robust privacy controls that allow users to manage how their data is used and stored.

2. Gather Your Resources (Define Clearly)

2.1. For SRS Documents

- Requirements Gathering Techniques
    - Interviews, surveys, and document analysis.

    - User stories and scenarios.
    - Brainstorming sessions.
- Functional and Non-Functional Requirements
    - Understanding the differences and examples of each.
    - Techniques for categorizing and documenting them.
- SRS Document Structure
    - Standard templates and formats.
    - Sections like introduction, overall description, and specific requirements.
- Quality Attributes in SRS
    - Scalability, maintainability, reliability, usability.
    - How to define and specify quality requirements.
- Validation and Verification of Requirements
    - Techniques to ensure requirements are complete, consistent, and testable.
    - Peer reviews, traceability, and requirements change management.

2.2. For UML Diagrams Documents

- Use Case Diagrams
    - Identifying actors and use cases.
    - Relationships like include, extend, and generalization.
- Activity Diagrams
    - Workflow and business process modeling.
    - Decision points, parallel processes, swimlanes.
- Sequence Diagrams
    - Object interactions over time.
    - Lifelines, messages, and sequence flow.
- Class Diagrams
    - Modeling structure and relationships of system classes.
    - Attributes, operations, and associations.
- State Diagrams
    - State transitions in a system.
    - Events, states, and actions.
- Component Diagrams
    - High-level architecture of a system.
    - Interfaces, components, and their relationships.
- Deployment Diagrams
    - Physical deployment of software on hardware.
    - Nodes, artifacts, and their association.

2.3. Additional Concepts:

- Modeling Standards and Notations
    - UML notation guidelines.
    - Best practices for clear and effective diagramming.
- Tools for UML Diagramming
    - Software like Lucidchart, StarUML, and Visio.
    - Features and choosing the right tool for your needs.
- Integrating SRS and UML
    - How UML diagrams support and enhance the SRS.
    - Consistency between requirements and diagrams.

3. Team Structure and Roles

- **Project Coordinator - Member 1:**
    - Oversee project management, organize meetings, maintain communication, and ensure project alignment.
    - **Skills Required:** Leadership, organizational, communication.
- **Requirements Analyst - Member 2:**
    - Gather and analyze requirements, conduct research, document and share requirements.
    - **Skills Required:** Analytical, attention to detail, interviewing, and documentation.
- **UML and System Design Specialist - Member 3:**
    - Develop UML diagrams, collaborate for system representation, and align diagrams with system requirements.
    - **Skills Required:** UML knowledge, design, technical understanding.
- **SRS Document Writer - Member 4:**
    - Draft the SRS document, incorporate requirements, and revise based on feedback.
    - **Skills Required:** Writing, technical knowledge, attention to detail.
- **Quality Assurance and Reviewer - Member 5:**
    - Review SRS documents and UML diagrams, provide feedback, and ensure quality standards.
    - **Skills Required:** Critical thinking, attention to detail, communication.

Start

4. Listing Functional Requirements

4.1. Gather Requirements:

Functional Requirements

- **User Account Management**
  - **User Registration:** Allow users to create an account.
  - **User Login/Logout:** Secure login and logout functionality.
  - **Account Recovery:** Enable password reset and account recovery options.
- **Facial Recognition for Mood Detection**
  - **Capture Image:** Allow users to capture a selfie or use a camera for real-time facial recognition.
  - **Mood Analysis:** Analyze the facial expressions to determine the user's mood.
- **Music Recommendation**
  - **Generate Playlists:** Suggest music playlists based on the detected mood.
  - **Custom Recommendations:** Tailor music suggestions based on the user’s past preferences and selections.
- **Mood Trend Analysis**
  - **Track Mood History:** Record and display the user’s mood over time.
  - **Provide Insights:** Offer insights or tips based on mood trends through news, stories, or exercises.
- **Integration with Music Libraries**
  - **Access to Music Database:** Connect with external music libraries, and streaming services, or create your own music
- **Data Management and Privacy Settings**
  - **Privacy Controls:** Let users manage their data privacy settings, especially regarding facial recognition data.
- **Feedback and Support**
  - **User Feedback:** Include a feature for users to give feedback on app functionality and music suggestions.
  - **In-App Support:** Provide a help or support section for user assistance.

4.2. Categorize and Prioritize

- **Essential (Must-Have)**

    - User Account Management
    - Facial Recognition for Mood Detection
    - Music Recommendation
    - Integration with Music Libraries
- **Important (Should-Have)**
    - Mood Trend Analysis
    - User Interface and Interaction
    - Settings and Customization
- **Optional (Nice-to-Have)**
    - Social Media Integration
    - In-App Purchases
    - Enhanced Accessibility Features

5. Defining High-Level Objects and Developing UML Diagrams

5.1. Identify Key Objects

- **User**
    - Attributes: UserID, Name, Email, Password, MoodHistory, UserPreferences
    - Methods: Login(), Logout(), UpdateProfile(), ViewMoodHistory().
- **MoodDetector**
    - Attributes: CameraInput, DetectedMood.
    - Methods: CaptureImage(), AnalyzeMood().
- **MusicRecommender**
    - Attributes: RecommendedPlaylist, UserPreferences.
    - Methods: GeneratePlaylist(Mood), UpdatePreferences(UserPreferences).
- **Playlist**
    - Attributes: PlaylistID, Songs, Genre.
    - Methods: CreatePlaylist(), AddSong(Song), RemoveSong(Song).
- **Song**
    - Attributes: SongID, Title, Artist, Genre.
    - Methods: Play(), Pause(), AddToPlaylist(Playlist).
- **MoodTrendAnalyzer**
    - Attributes: MoodData, User.
    - Methods: AnalyzeTrends(), GenerateReport().

Tips for Identifying Objects

- Focus on the main functionalities of your app and the data it needs to handle.

- Consider objects not only as tangible entities but also as conceptual parts of your system (like **MoodDetector** or **MoodTrendAnalyzer**).
- Avoid overcomplicating the object model. Start with the most critical objects and add more as needed.

5.2. Develop UML Diagrams

5.2.1. Use Case Diagram

- Actors: User, Music.
- **UC User**
    - Registering for a New Account
        - Users can create a new account by providing required details such as email, username, and password.
    - Logging into the App
        - Users can log into their account using their username and password.
    - Recovering a Forgotten Password
        - Users can recover their password through a 'forgot password' feature, typically involving email verification.
    - Capture Mood Using Facial Recognition
        - Users can have their mood detected through the app's facial recognition feature, which analyzes their facial expressions.
    - Receive Music Recommendations
        - Based on the detected mood, the app suggests a selection of music that aligns with the user's current emotional state.
    - Customize Music Preferences
        - Users can customize their music preferences, allowing the app to provide more tailored suggestions over time.
    - View Mood History
        - Users can view their past mood detections and the music played during those times.
    - Manage Account Settings
        - Users can manage their account settings, including personal information, password, and other preferences.
- **UC Music Streaming Service**
    - Provide Music Tracks and Playlists
        - The music streaming service integrated with the app provides a vast selection of music tracks and playlists for recommendation.

- Update Music Library
    - The service regularly updates its music library to include new tracks and playlists, ensuring that users have access to a wide range of music.

5.2.2. Class Diagram

- **Class: User**
    - Attributes:
        - UserID (String)
        - Name (String)
        - Email (String)
        - Password (String)
        - MoodHistory (List<MoodRecord>)
    - Methods:
        - Login(email: String, password: String): Boolean
        - Logout(): Void
        - UpdatePreferences(settings: UserSettings): Void
        - ViewMoodHistory(): List<MoodRecord>
    - Relationships:
        - Interacts with MoodDetector and MusicRecommender.
- **Class: MoodDetector**
    - Attributes:
        - CameraInput (CameraStream)
        - DetectedMood (MoodType)
    - Methods:
        - CaptureImage(): Image
        - AnalyzeMood(image: Image): MoodType
    - Relationships:
        - Provides DetectedMood to MusicRecommender.
- **Class: MusicRecommender**
    - Attributes:
        - UserPreferences (UserSettings)
        - RecommendedPlaylist (Playlist)
    - Methods:
        - GeneratePlaylist(mood: MoodType): Playlist
        - UpdatePreferences(preferences: UserSettings): Void
    - Relationships:
        - Uses DetectedMood from MoodDetector to generate playlists.
        - Composes Playlist objects.

- **Class: Playlist**
    - Attributes:
        - PlaylistID (String)
        - Songs (List<Song>)
        - MoodType (MoodType)
    - Methods:
        - AddSong(song: Song): Void
        - RemoveSong(song: Song): Void
    - Relationships:
        - Contains multiple Song objects.
- **Class: Song**
    - Attributes:
        - SongID (String)
        - Title (String)
        - Artist (String)
        - Duration (Time)
    - Methods:
        - Play(): Void
        - Pause(): Void
    - Relationships:
        - Part of Playlist objects.

5.2.3. Activity Diagrams

- **User Registration and Login Process**
    - Start with the user opening the app.
    - Select the option to register or log in.
    - Enter registration/login details.
    - Verification of user credentials.
    - Access is granted to the main app interface.
- **Mood Detection Process**
    - The user initiates a mood detection feature.
    - The camera captures the user's facial expression.
    - Image processing and mood analysis.
    - The mood detection result is displayed.
- **Music Recommendation Process**
    - The system receives mood data from MoodDetector.
    - Fetch music preferences from the user profile.
    - MusicRecommender generates a playlist based on mood.
    - The playlist is displayed to the user.
- **Mood History Review Process**

    - The user navigates to the mood history section.
    - The system retrieves mood history data.
    - Display mood trends and historical data.
    - The user interacts with mood history (e.g., selects a specific date).
- **User Profile Management**
    - The user accesses profile settings.
    - Options to update personal info, change passwords, and music preferences.
    - Submit changes.
    - System updates user profile.
- **Feedback and Support Process**
    - The user selects the feedback/support option.
    - Fill in and submit a feedback form or support query.
    - System logs feedback/query.
    - Automated or manual response process initiated.
- **Playlist Customization Process**
    - The user selects a recommended playlist.
    - Option to add or remove songs.
    - Save changes to the playlist.
    - The system updates the playlist accordingly.
- **Logging Out/Exiting the App**
    - The user chooses to log out or exit the app.
    - The system performs logout procedures.
    - Closure of user session.
    - Return to the login screen or app closure.

5.2.4. Sequence Diagrams

User Registration Sequence

- Objects: User Interface (UI), User, Registration System.
- Flow:
    - The user accesses the UI and selects 'Register'.
    - The user enters details (username, email, password) on the UI.
    - UI sends registration details to the Registration System.
    - Registration System validates and creates a new user account.
    - Registration System sends a confirmation or error message back to the UI, which is then displayed to the User.

User Login Sequence

- Objects: User Interface, User, Login System.
- Flow:
    - The user enters login credentials on the UI.

- UI sends these credentials to the Login System.
- Login System verifies credentials.
- Login System sends a success or failure response to the UI.
- The UI updates the User with the login status.

Mood Detection Sequence

- Objects: User, MoodDetector, Camera System.
- Flow:
  - The user initiates mood detection on the UI.
  - Camera System activates and captures the user's facial expression.
  - Camera System sends the image to the MoodDetector.
  - MoodDetector analyzes the image and determines the mood.
  - MoodDetector returns the mood result to the UI for the User to view.

Music Recommendation Sequence

- Objects: User, MusicRecommender, Music Database.
- Flow:
  - The user's detected mood is sent to the MusicRecommender.
  - MusicRecommender queries the Music Database based on the mood.
  - Music Database returns a list of suitable tracks/playlists.
  - MusicRecommender sends these recommendations back to the UI for the User.

Playlist Creation Sequence

- Objects: User, Playlist Manager, Music Database.
- Flow:
  - The user selects tracks and requests a new playlist via the UI.
  - Playlist Manager receives the request and interacts with the Music Database to gather the tracks.
  - Music Database sends the tracks to the Playlist Manager.
  - Playlist Manager creates the playlist and confirms back to the User through the UI.

Mood History Review Sequence

- Objects: User, UI, Mood History Manager.
- Flow:
  - The user requests to view mood history on the UI.
  - Mood History Manager retrieves the user’s mood history data.
  - Mood history is displayed to the User via the UI.

Account Settings Update Sequence

- Objects: User, UI, Account Settings Manager.
- Flow:
    - The user accesses settings through the UI and makes changes.
    - UI sends the updated settings to the Account Settings Manager.
    - Account Settings Manager updates the user's profile and confirms the update.
    - UI notifies the User of the successful update.

Feedback Submission Sequence

- Objects: User, Feedback System.
- Flow:
    - The user submits feedback through the UI.
    - Feedback is sent to the Feedback System.
    - Feedback System stores the feedback and optionally sends a confirmation or thank you message to the User.

6. Regular Team Meetings and Minutes

The following is the project plan. Based on this plan, you will know how to plan your team to finish the assignment step by step and record regular meetings.

Project Plan for SRS and UML Overview Development

Project Duration: Assume a duration of 3 months for this phase.

Team Structure:

- Project Coordinator
- SRS Document Writer
- UML Specialist
- QA/Reviewer

Monthly Breakdown:

Month 1: SRS Development

- Week 1 Meeting: Kick-off and Requirement Analysis
    - Discuss the project scope and detailed requirements.
    - Assign Tasks:
        - SRS Writer to draft initial sections of the SRS document.
        - UML Specialist to begin identifying key system components.
    - Meeting Minutes: Record decisions made, assigned tasks, and deadlines.
- Week 2-4 Meetings: SRS Progress Check
    - Weekly meetings to review the progress of the SRS document.
    - Update Tasks:

        - Review and refine sections of the SRS.
        - Discuss any new requirements identified.
    - Meeting Minutes: Note updates, any changes in requirements, and additional tasks assigned.

Month 2: UML Development

- Week 5 Meeting: UML Diagram Planning
    - Outline the required UML diagrams (Use Case, Class, Activity, Sequence Diagrams).
    - Assign Tasks:
        - UML Specialist to start creating initial diagrams.
    - Meeting Minutes: Record diagram types agreed upon, and tasks assigned.
- Week 6-8 Meetings: UML Diagram Review
    - Weekly review of UML diagrams.
    - Update Tasks:
        - Refine UML diagrams based on team feedback.
    - Meeting Minutes: Track the progress of diagrams, capture feedback, and note any revisions needed.

Month 3: Review and Finalization

- Week 9-10 Meeting: Finalizing SRS and UML Diagrams
    - Review the final draft of the SRS and UML diagrams.
    - Assign Tasks:
        - Final edits and refinements based on the team's feedback.
    - Meeting Minutes: Document final changes needed, and ensure all team members are aligned.
- Week 11 Meeting: Quality Assurance
    - QA/Reviewer to ensure all specifications and diagrams meet the required standards.
    - Meeting Minutes: Record any QA feedback, and final adjustments needed.
- Week 12 Meeting: Final Review and Submission Preparation
    - Final review of SRS and UML diagrams.
    - Prepare documents for submission.
    - Meeting Minutes: Confirm completion of all sections, and document preparation for submission.

7. Finalizing and Review

Review and Edit

- Content Review: Check all written content for clarity and accuracy. Ensure that the SRS document accurately describes functionalities like emoji selection, music integration, user interface, and any social sharing features.
- Technical Accuracy: Confirm that the UML diagrams correctly represent the app's architecture, including classes like User, MusicTrack, Emoji, and SocialShare.
- Usability Considerations: Verify that the app's design considerations, such as ease of emoji selection and music navigation, are well-documented.

Peer Review

- Functionality Check: Team members can use scenarios to test if the app's described features (like selecting an emoji for a specific song or sharing emojis on social media) align with the SRS.
- Interface Review: Ensure that the design elements (like the emoji palette layout or the music player interface) are intuitive and user-friendly.
- Feedback Incorporation: Use team feedback to refine and improve the SRS and UML diagrams, focusing on user experience and technical feasibility.

Finalize Documentation

- Consistency Check: Ensure that the documentation has a consistent format, and technical terms are used uniformly.
- Final Edits: Make any necessary last-minute edits to improve clarity and conciseness.
- Documentation Package: Prepare the final package of the SRS document along with UML diagrams, ensuring all components are well-organized and easily navigable.

8. Presentation and Submission

Prepare Presentation

- Summarize Key Points: Create slides summarizing the app's main features, such as emoji-music integration, custom emoji creation, and social sharing functionalities.
- Visual Aids: Include screenshots of UML diagrams and potential app interfaces to give a clear visual understanding.
- Narrative Flow: Develop a clear storyline for your presentation, beginning with the app's purpose, user journey, and key features, and concluding with its technical architecture.

Practice and Feedback

- Mock Presentations: Conduct practice runs of the presentation with your team to ensure smooth delivery.
- Incorporate Feedback: Gather feedback from these sessions to refine your presentation, focusing on clarity, pacing, and engagement with the audience.
- Confidence Building: Use these practice sessions to build confidence in presenting technical details and answering potential questions.

Submit Assignment

- Final Check: Do a final review of all documents and presentation materials for completeness and accuracy.
- Packaging for Submission: Organize your SRS document, UML diagrams, and presentation slides in the required format (like PDF, PPT, etc.).
- Timely Submission: Submit your assignment before the deadline, ensuring that all components are included.