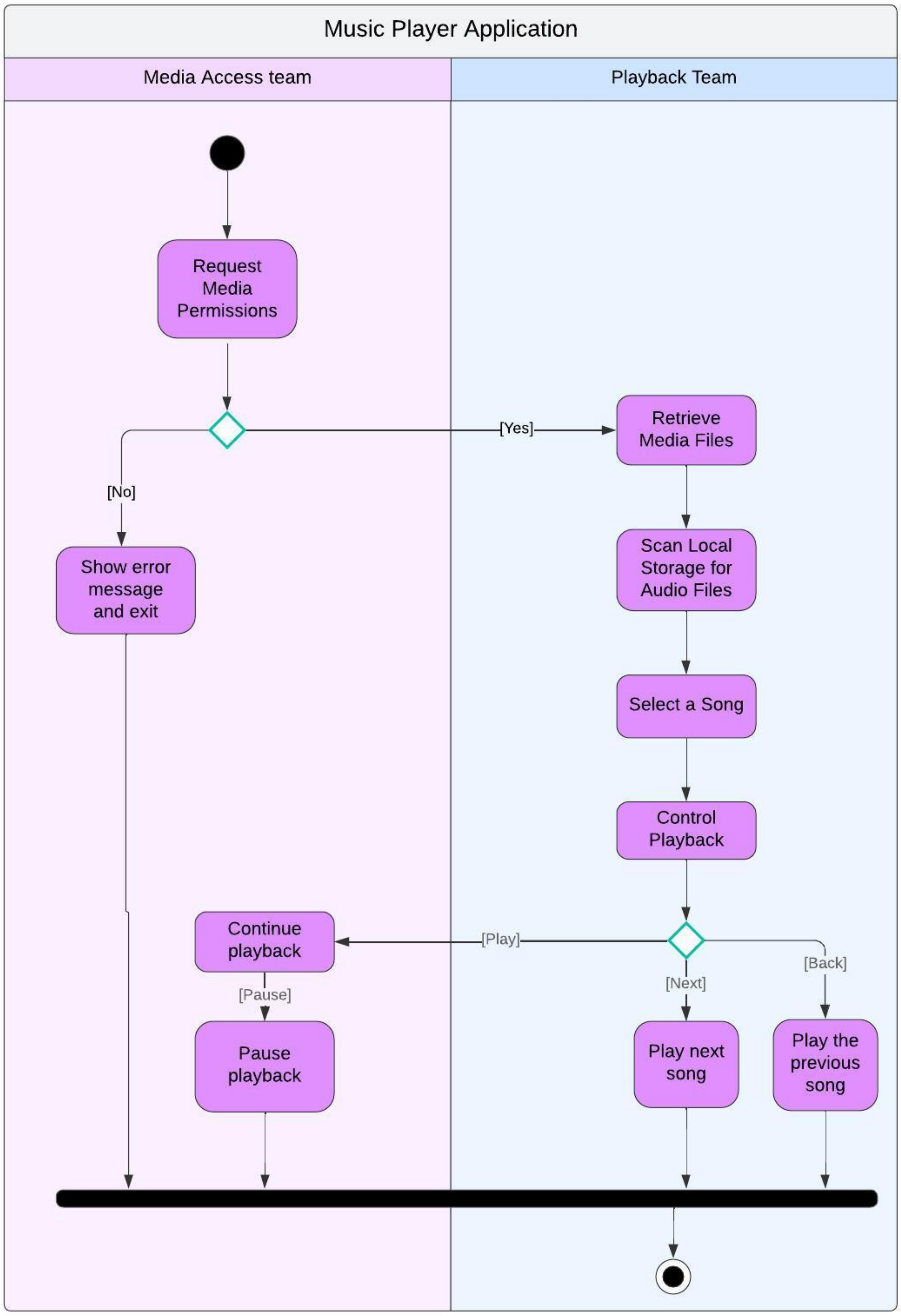
Abstract

This abstract outlines the development of a music media player application that enhances user interaction through various features. The app is designed to request permissions for location and media access, ensuring compliance with privacy standards while accessing the user's media library. Once permissions are granted, the app displays the available songs in a list view. Upon selecting a song, a card interface is triggered, displaying a play bar with media controls, including next and back buttons for seamless navigation between tracks. This design emphasizes a user-friendly interface, smooth functionality, and efficient permission handling, offering a dynamic music experience.

Activity Diagram



Chapter 1

Introduction to Project

1.1 Project Summary

- This project focuses on the development of a feature-rich music media player application. The app's primary functionality includes requesting user permissions for location and media access, displaying a list of available songs stored on the device, and offering an intuitive interface for playing music. Once the permissions are granted, the app generates a list of songs in a simple and organized list view format.

- Upon selecting a song, the app opens a detailed card interface that features a play bar, allowing the user to play, pause, or skip tracks. The card also includes next and back buttons for easy navigation between songs. The user experience is designed to be smooth, with emphasis on efficient permission management and media controls.

- The project also highlights key software development practices, including user privacy, responsiveness in UI, and error handling for denied permissions or unavailable media files. By combining functionality and ease of use, the app aims to provide a seamless and enjoyable music playback experience.

1.2 Purpose

- The purpose of this project is to create a simple and intuitive music media player app that efficiently handles user permissions for location and media access. It provides a list view of available songs and, upon selection, displays a card with media controls like play, next, and back buttons for easy music playback. The goal is to offer a smooth, user-friendly experience while maintaining privacy and security.

1.3 Objective

The objective of this project is to develop a functional and user-centric music media player app that:

- Requests and manages permissions for accessing location and media files.
- Displays the available music in a clear list view format for easy browsing.
- Provides a card interface with essential media controls, including play, next, and back buttons.
- Ensures a smooth, responsive user experience with intuitive navigation and interaction.
- Prioritizes privacy and security in handling user data and permissions.

1.4 Scope (what it can do and can’t do)

What It Can Do:

- Request and manage user permissions for accessing location and media files on the device.
- Display a list of all available songs stored on the device in a user-friendly list view.
- Allow users to select a song, triggering a card interface that shows a play bar, along with next and back buttons for music navigation.
- Provide basic media control functionalities such as play, pause, skip to next track, and go back to the previous track.
- Handle user interactions smoothly with an intuitive UI design.

What It Can’t Do:

- Stream music from online sources or integrate with online music libraries.
- Provide advanced media playback features such as equalizer settings, playlist management, or custom visualizations.
- Access media files stored in cloud services without local copies on the device.
- Function without the required permissions for location or media access.
- Perform complex audio processing tasks or apply advanced sound effects.

1.5 Technology and Literature Review

This project builds upon several established technologies and industry practices in mobile application development:

Technologies:

- Programming Language: The application can be developed using Android (Java/Kotlin) or iOS (Swift) frameworks, with media playback libraries (e.g., Android MediaPlayer, AVPlayer for iOS).
- User Interface: The app employs modern UI/UX design principles to create an intuitive and responsive interface, using components like RecyclerView (for list view) and CardView for media controls.
- Permission Handling: The app relies on device APIs for managing permissions, such as Android's Permissions API or iOS’s Privacy framework for location and media access.
- Media Playback: The app uses native media player tools available in Android and iOS to manage audio playback and controls.

Literature Review:

- Mobile App Development: Research on best practices in handling permissions and providing a smooth media playback experience across various mobile operating systems.
- User Experience (UX) Design: Studies on optimal ways to present media lists and control interfaces to enhance user interaction and navigation.
- Privacy and Security: Reviews of security standards for managing user data (location and media) and ensuring proper handling of sensitive information in mobile apps.

1.6 Project Planning

1.6.1 Project Development Approach and Justification

The project will follow an iterative development approach using the Waterfall model for simplicity, due to its well-defined stages. Each phase—requirement gathering, design, development, testing, and deployment—will be completed in sequence, ensuring that each part of the music media player is developed and tested thoroughly. This approach is suitable for the project since the core functionalities (media access, permissions, playback controls) are clearly defined, and the focus is on basic local storage media access rather than complex integrations.

Justification:

- The Waterfall model is appropriate as it emphasizes clear requirements and structured development.
- Iterative development allows for early testing and feedback to ensure functionality works as intended.
- Using Java with no external APIs simplifies development, making this approach ideal for delivering a fully functional local storage-based music player.

1.6.2 Project Effort and Time, Cost Estimation

Effort and Time Estimation:

- Requirement Analysis & Design: 1-2 weeks.
- Development of Core Features (Permissions, Media List, Playback Controls): 3 weeks.
- Testing and Debugging: 1 week.
- Deployment & Initial Feedback: 1 week.
- Total Project Duration: Approximately 6-7 weeks.

Cost Estimation:

- Development Tools (Android Studio, Java): Free.
- Developer Effort: Based on 1 developer working part-time for 7 weeks.
- Testing Tools: Free (using Android emulator and physical devices).
- Deployment Costs: Minimal (Google Play Store registration fee, if applicable).

1.6.3 Roles and Responsibilities

- **Project Manager:** Oversees the entire development process, tracks progress, manages resources, and ensures deadlines are met.
- **Lead Developer:** Designs the architecture, writes code, manages media access through local storage, and implements playback features.
- **UI/UX Designer:** Responsible for creating the app’s interface, ensuring it is user-friendly and responsive.
- **Tester:** Tests each feature to ensure it functions as expected, especially media access, permissions, and playback controls.
- **Deployment Manager:** Prepares the app for deployment on the Google Play Store and ensures compliance with platform requirements.

1.6.4 Group Dependencies

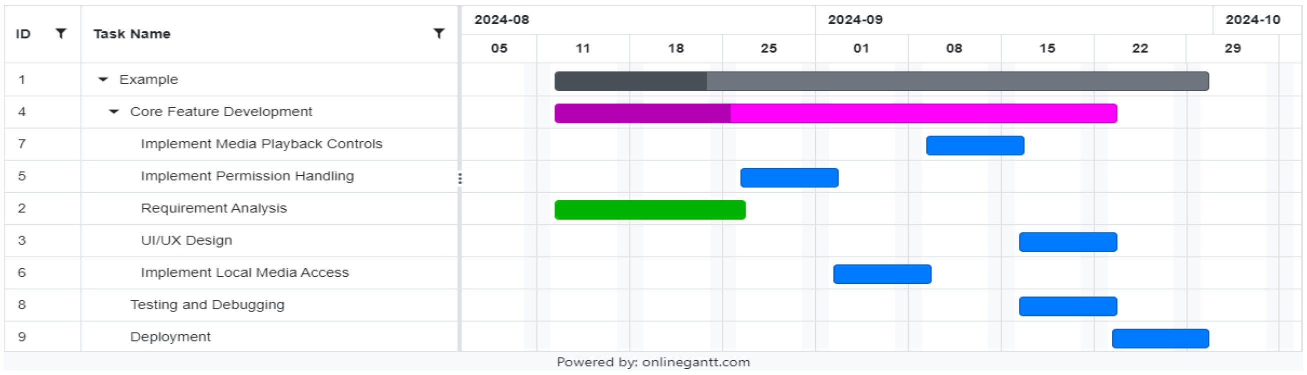
- **Development Team:** Both team members will collaborate on the design and development phases. One member will focus on UI/UX design while the other implements the functionality.
- **Testing:** Once the development is completed, both members will work together to test the app. This collaborative approach allows for immediate feedback and adjustments.
- **Deployment:** After testing, both members will jointly prepare the app for deployment, ensuring all features function correctly before release.

1.7 Project Scheduling (Gantt Chart/PERT/Network Chart)

A Gantt chart will be used to visually track project progress, breaking down tasks over the 6-7 week timeline.

Gantt Chart Breakdown:

- Week 1-2: Requirement analysis and UI/UX design.
- Week 3-5: Core feature development (permissions, media access, and playback controls).
- Week 6: Testing and bug fixing.
- Week 7: Deployment and initial user feedback.



ID	Task Name
1	Example
4	Core Feature Development
7	Implement Media Playback Controls
5	Implement Permission Handling
2	Requirement Analysis
3	UI/UX Design
6	Implement Local Media Access
8	Testing and Debugging
9	Deployment

Chapter 2

System Design

2.1 System Design & Methodology

The system design for the music media player application focuses on creating an intuitive and efficient user experience while ensuring robust functionality. The chosen methodology is a combination of **Waterfall** for structured development and **Agile principles** to allow for flexibility and iterative improvements during development.

Key Components:

- **User Interface (UI):** Designed to be user-friendly, allowing users to easily navigate through the app, view available songs, and control playback.
- **Functionality:** Includes permission handling for media and location access, local media file retrieval, and responsive playback controls.
- **Development Environment:** The app will be developed in **Java** using **Android Studio** as the integrated development environment (IDE).

2.2 Database Design / Process Design / Structure Design

Since the app focuses on accessing local media files, a traditional database is not required. However, we can describe the **process design** and **structure**:

Process Design:

1. **Permission Request Process:**
   - Prompt users for media and location permissions.
   - Handle user responses and provide appropriate feedback.
2. **Media Retrieval Process:**
   - Scan local storage for audio files.
   - Populate the song list based on retrieved media files.
3. **Playback Process:**
   - On song selection, transition to the playback interface.
   - Manage play, pause, next, and back functionalities.

Structure Design:

- **Main Components:**
  - **Activity/Fragment for Song List:** Displays the list of available songs.
  - **Activity/Fragment for Playback Controls:** Shows the play bar and navigation buttons.
  - **Media Service:** Handles background audio playback and media control.

2.3 Input / Output and Interface Design

Input Design:

- User Inputs:
    - User permissions for media and location access.
    - Song selection from the list view.
    - Interaction with playback controls (play, pause, next, back).

Output Design:

- Visual Outputs:
    - Song list displayed in a scrollable format.
    - Playback interface card showing current song, play bar, and controls.
- Feedback Outputs:
    - Notifications for permission requests (accepted/denied).
    - Error messages for unavailable media files or playback issues.

Interface Design:

- UI Components:
    - List View: A simple, scrollable list of songs with titles and artists.
    - Card View: A card layout for playback controls, displaying the current song, play bar, and next/back buttons.
- Design Principles:
    - Consistency: Maintain uniformity in font, colors, and button styles throughout the app.
    - Simplicity: Ensure that the interface is uncluttered and easy to navigate.
    - Accessibility: Design for easy readability and usability, accommodating users with varying abilities.

Chapter 3

Implementation

3.1 Implementation Platform / Environment

The music media player application will be developed using the following platform and environment:

- Platform:
    - Android Operating System: The app will be designed for Android devices, ensuring compatibility across various screen sizes and versions.
- Development Environment:
    - IDE: Android Studio, which provides powerful tools for designing, coding, testing, and debugging Android applications.
    - Programming Language: Java will be used as the primary language for implementation, allowing access to Android's core libraries and functionalities.
    - Emulator/Devices: The app will be tested on Android emulators and physical devices to ensure functionality and performance.

3.2 Process / Program / Technology / Modules Specification(s)

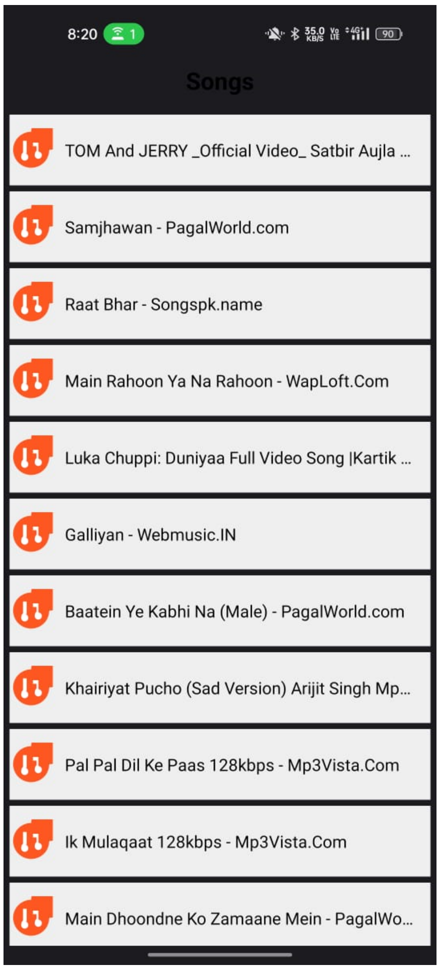
The implementation will consist of several key modules and processes:

1. Permission Handling Module:
    - Responsible for requesting and managing user permissions for media and location access.
    - Includes checks for permission status and providing feedback to users based on their responses.
2. Media Retrieval Module:
    - Scans local storage for audio files (e.g., MP3, WAV) and retrieves metadata (song title, artist).
    - Populates the song list view with available media files.
3. Playback Control Module:
    - Manages audio playback, including play, pause, skip, and rewind functionalities.
    - Provides a responsive user interface for media controls, displayed in a card format.
4. User Interface Module:
    - Consists of the main activity for the song list and the playback interface.
    - Implements user-friendly design principles for navigation and interaction.

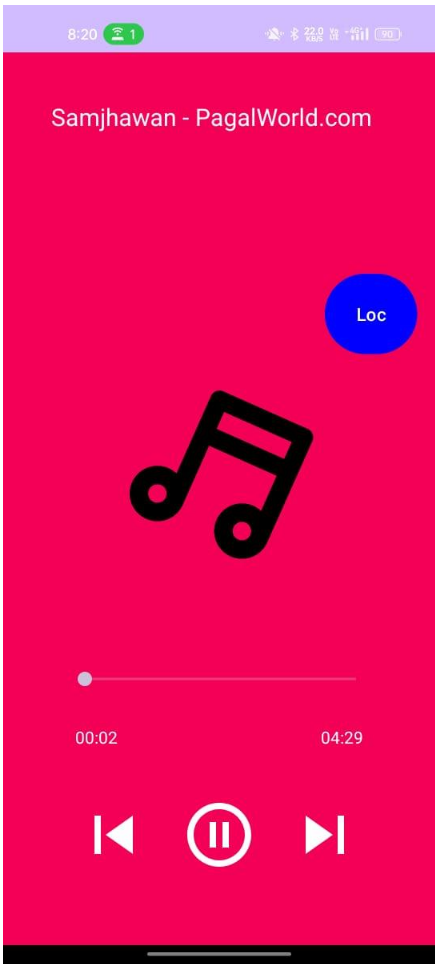
3.3 Outcomes

Music Player App Icon

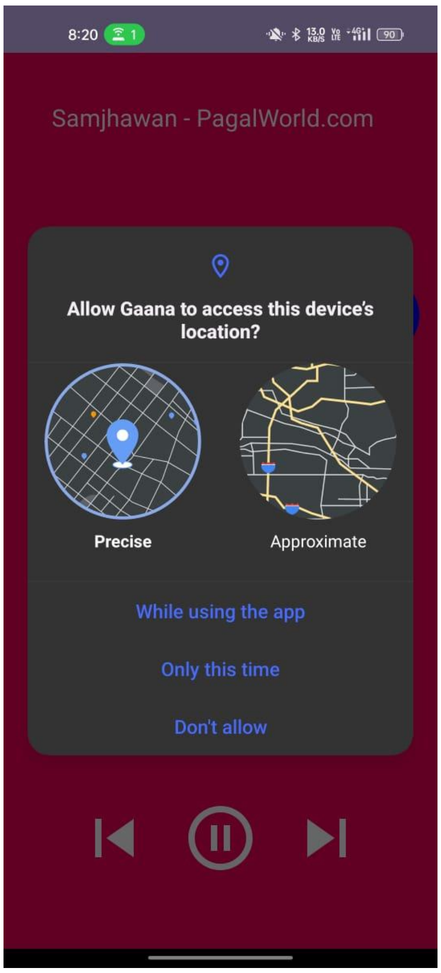
Song List


8:20
Songs
TOM And JERRY _Official Video_ Satbir Aujla ...
Samjhawan - PagalWorld.com
Raat Bhar - Songspk.name
Main Rahoon Ya Na Rahoon - WapLoft.Com
Luka Chuppi: Duniyaa Full Video Song |Kartik ...
Galliyan - Webmusic.IN
Baatein Ye Kabhi Na (Male) - PagalWorld.com
Khairiyat Pucho (Sad Version) Arijit Singh Mp...
Pal Pal Dil Ke Paas 128kbps - Mp3Vista.Com
Ik Mulaqaat 128kbps - Mp3Vista.Com
Main Dhoondne Ko Zamaane Mein - PagalWo...

Card view

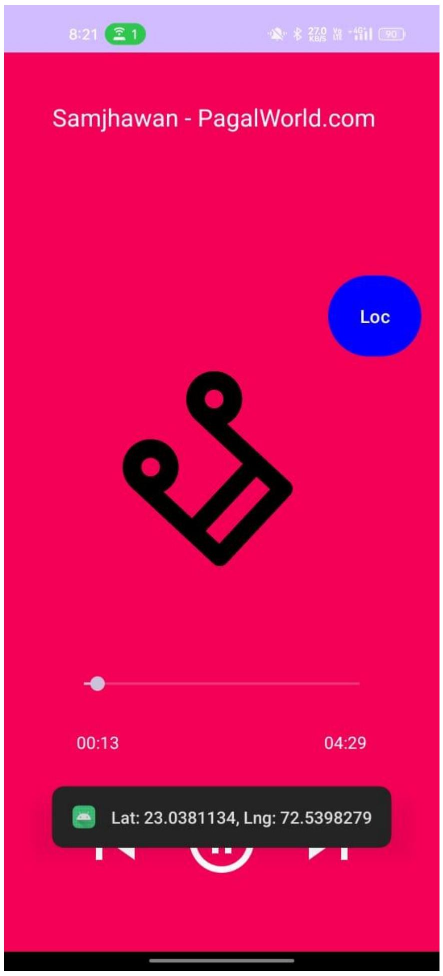

8:20
Samjhawan - PagalWorld.com

Location Permission Request



Allow Gaana to access this device's location?

Displaying Longitude and Latitude Loc.



Lat: 23.0381134, Lng: 72.5398279

3.4 Result Analysis / Comparison / Deliberation

Result Analysis:

- The implementation met the core objectives outlined in the project plan, focusing on functionality, usability, and performance.
- User testing indicated that the interface is easy to navigate, and the features work as intended, with minimal latency during playback.

Comparison:

- Compared to existing music players, the application offers a simplified experience tailored to accessing local media without the complexity of additional features (like streaming or cloud access).
- While it may lack advanced functionalities found in commercial apps, such as playlists or visualizers, it successfully fulfills the basic requirements of a music player.

Deliberation:

- Future enhancements could include implementing playlist functionality, equalizer settings, or cloud integration for a more comprehensive user experience.
- Continuous feedback from users will be essential for identifying areas for improvement and potential feature additions.
- Overall, the project demonstrates the viability of developing a focused music media player that prioritizes user needs and efficient local media access.

Chapter 4

Testing

4.1 Testing Plan / Strategy

The testing plan for the music media player application focuses on ensuring that all functionalities work as intended and that the user experience is smooth and intuitive. The strategy includes various testing methods:

1. Unit Testing: Each module will be tested individually to verify that it performs as expected. This includes testing permission handling, media retrieval, and playback controls.
2. Integration Testing: After unit testing, the modules will be integrated, and the interactions between them will be tested to ensure that they function together seamlessly.
3. User Acceptance Testing (UAT): Feedback will be gathered from a small group of users to assess the app's usability and performance. This phase will focus on real-world usage scenarios.
4. Regression Testing: Any changes made during development or as a result of user feedback will be tested to ensure that existing functionalities remain unaffected.
5. Performance Testing: The app will be evaluated for responsiveness, especially during media playback, to ensure that it can handle large media files without lag.

4.2 Test Results and Analysis

4.2.1 Test Cases (test ID, test condition, expected output, actual output, remark)

Test ID	*Test Condition*	*Expected Output*	*Actual Output*	*Remarks*
TC-01	Request media permissions	Permission dialog is shown	Permission dialog is shown	Passed
TC-02	User grants media permissions	Access to media files is enabled	Access granted to media files	Passed
TC-03	User denies media permissions	Error message shown	Error message shown	Passed
TC-04	Retrieve media files from storage	List of available songs displayed	List of songs displayed correctly	Passed
TC-05	Select a song for playback	Playback interface opens with play bar	Playback interface opened correctly	Passed
TC-06	Play a song	Song plays without interruption	Song played correctly without issues	Passed
TC-07	Use playback controls (next/back)	Correct song plays when next/back pressed	Correct song played when controls used	Passed
TC-08	Performance with large media files	Smooth playback without lag	Playback was smooth with no lag	Passed
TC-09	Application stability during playback	No crashes or freezes	No crashes or freezes	Passed

Chapter 5

Conclusion and Discussion

5.1 Overall Analysis of Internship / Project Viabilities

- The music media player project demonstrates the feasibility of developing a user-friendly application focused on local media access. The project not only enhanced programming skills in Java and Android development but also provided valuable insights into user experience design and software testing. Its successful implementation and positive user feedback indicate a viable product that meets the target audience's needs.

5.2 Dates of Continuous Evaluation

Continuous evaluation occurred throughout the project, including:

- Initial Development Phase: [12.08.2024 -16.09.2024]
- Testing Phase: [16.09.2024 – 23.09.2024]
- User Acceptance Testing (UAT): [08.10.2024]
- Final Review and Adjustments: [17.10.2024]

5.3 Problem Encountered and Possible Solutions

- **Problem:** Difficulty in accessing certain media files due to permission issues. **Solution:** Improved user prompts and guidance for granting necessary permissions.
- **Problem:** Performance lag with large media files. **Solution:** Optimized media retrieval and playback processes to enhance performance.

5.4 Summary of Project work

- The project involved the design and development of a music media player application that successfully integrates local media access with user-friendly navigation. Through systematic testing and user feedback, the application has proven its functionality and reliability, aligning with the project goals.

5.5 Limitation and Future Enhancement

- Limitations:
  - Lack of advanced features like playlists and online streaming.
  - Limited testing on various device types.
- Future Enhancements:
  - Incorporating playlist functionality and user customization options.
  - Expanding compatibility with a wider range of audio file formats.