Industrial Internship Report on

"SomeGeetha-Music player application interface prototype based on core Java"

Prepared by

G. Namitha reddy

Indu priya shirisha

N.C.Supritha

| *Executive Summary* |
|---|
| This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

 This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time.

 My project was **"SomeGeetha music player application interface based on core java"**

 This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship. |

TABLE OF CONTENTS

1 Preface

Relevant internships are crucial for building one's career as they provide practice experience, industry exposure, and the opportunity to develop essential skills. They bridge the gap between theoretical knowledge and real-world application, enhancing employability and opening doors to future job prospects.
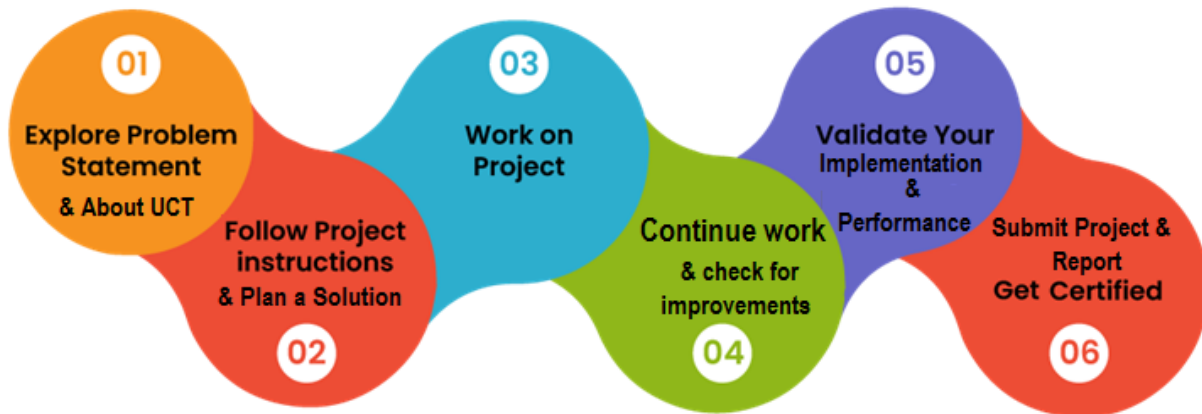
Summary of 6 weeks music player application

- **Week1:** Learned about UniConverge Technologies and IoT academy.

    The first progress was to form a team to complete this project. And we had a team consisting of four members. We discussed about the internship program and the industry partner.

    To explore the projects given to us and to plan accordingly for coming weeks.
- **Week2**: assigned a project to do.

    Learned many basic requirements to meet the project.

    Learned about many libraries in java.
- **Week3**: Started understanding writing of code for interference.

    Referred many GitHub codes to develop interface.
- **Week4:** Learned about GitHub and LinkedIn.

    Trying to reach the basic requirements of the application.

    Implemented the all the basic operation of application like pause, stop, play, duration.
- **Week5**: Understand what makes difference from other application, and tried to implement it.

- Add the feature like album, artist, favorites (add/remove from source library) and playlists (add/remove from playlist, create/remove playlists) etc.

- **Week6:** Completed project with all requirements.

    Add the project into GitHub.

➢ Gratitude towards internship provider: Being selected for an internship under USC/UCT is a significant opportunity. It provides a unique chance to work alongside esteemed faculty members, access advanced resources and facilities, and contribute to groundbreaking research. This internship enables students to enhance their knowledge, develop critical skills, and establish valuable connections in their field of interest. It is a prestigious opportunity that adds credibility to one's academic and professional profile, paving the way for future success and our career achievements.

➢ This project provided valuable learnings and an enriching overall experience. I gained practical knowledge in designing and developing a Music player application, utilizing programming languages, and tools. I enhanced my problem-solving skills, collaboration abilities, and project management capabilities. The project allowed me to understand the complexities of making separate playlist, import of local and online files, and user interface design. Overall, this project deepened my technical skills, expanded my understanding of the Music player, and provided a hands-on learning experience that will benefit my future endeavors.

➢ I would like to express my sincere gratitude to "Kaushlendra Singh Sisodia (Founder)" and the entire back-end team for their invaluable support and contributions throughout this project. The expertise and dedication have been instrumental in the successful implementation of the system. Thank you for your collaboration and assistance.

2   Introduction

2.1   About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.


i. UCT IoT Platform (uct Insight)

UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA

- It supports both cloud and on-premises deployments.

It has features to
- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine
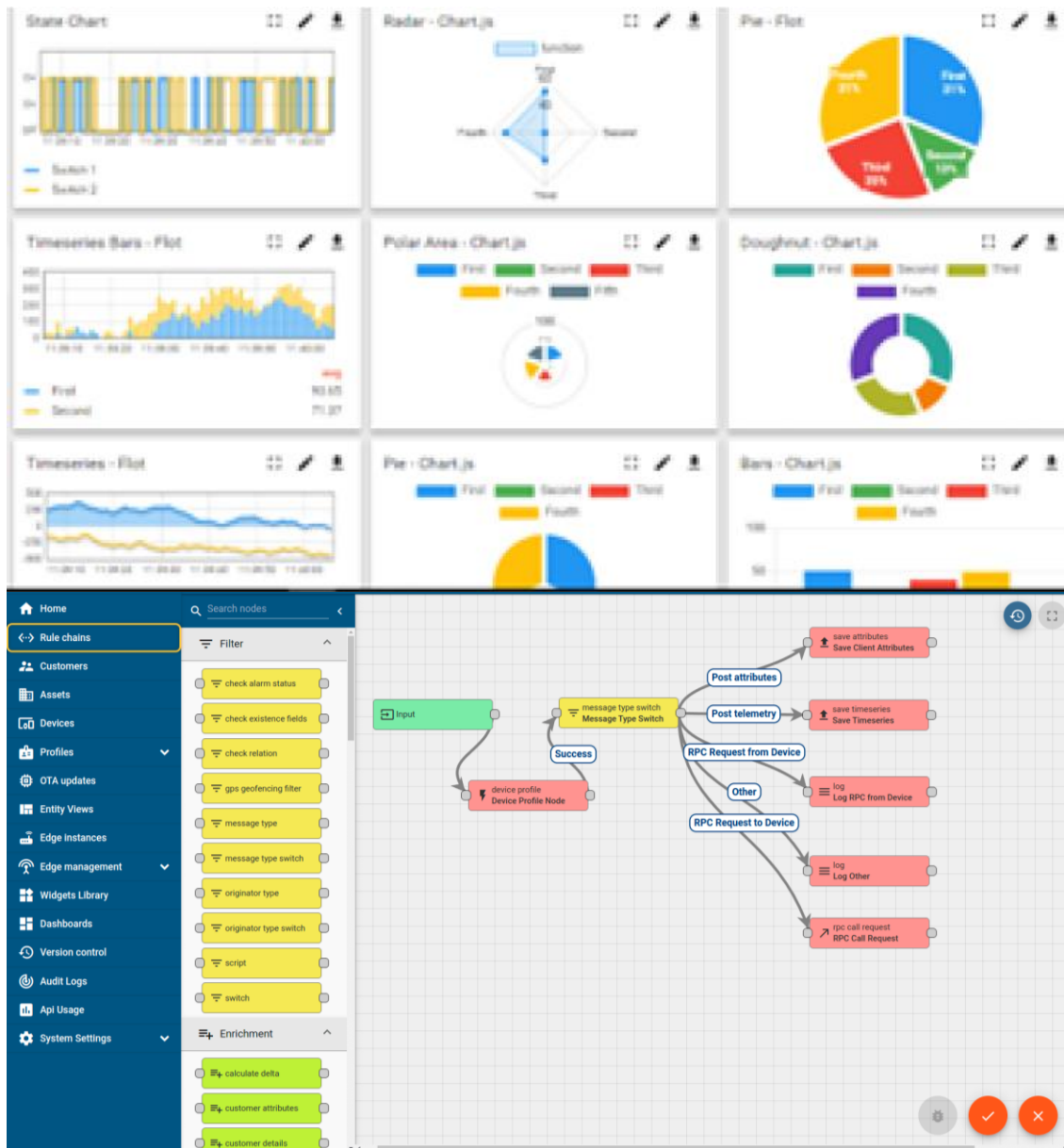
## ii. **Smart Factory Platform (** FACTORY WATCH **)**

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring

- OEE and predictive maintenance solution scaling up to digital twin for your assets.

- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.

- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

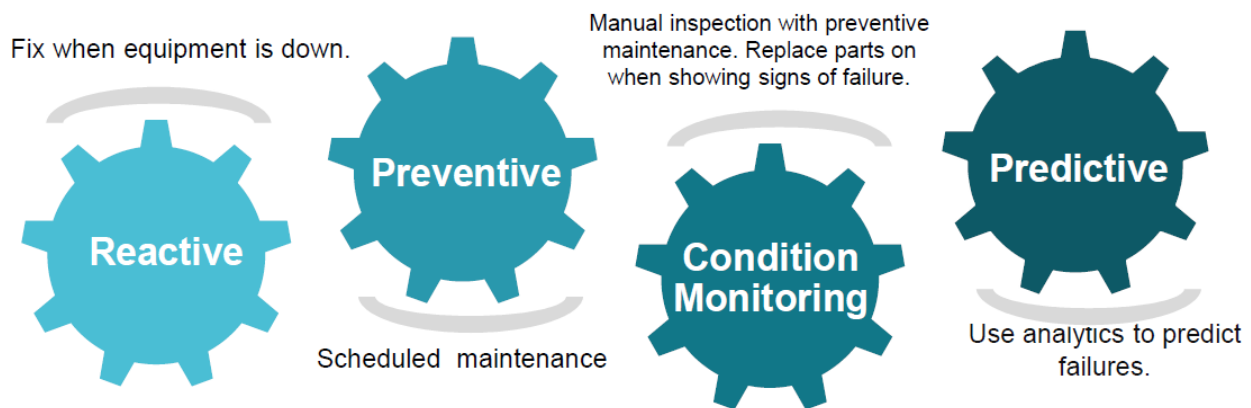Its unique SaaS model helps users to save time, cost and money.
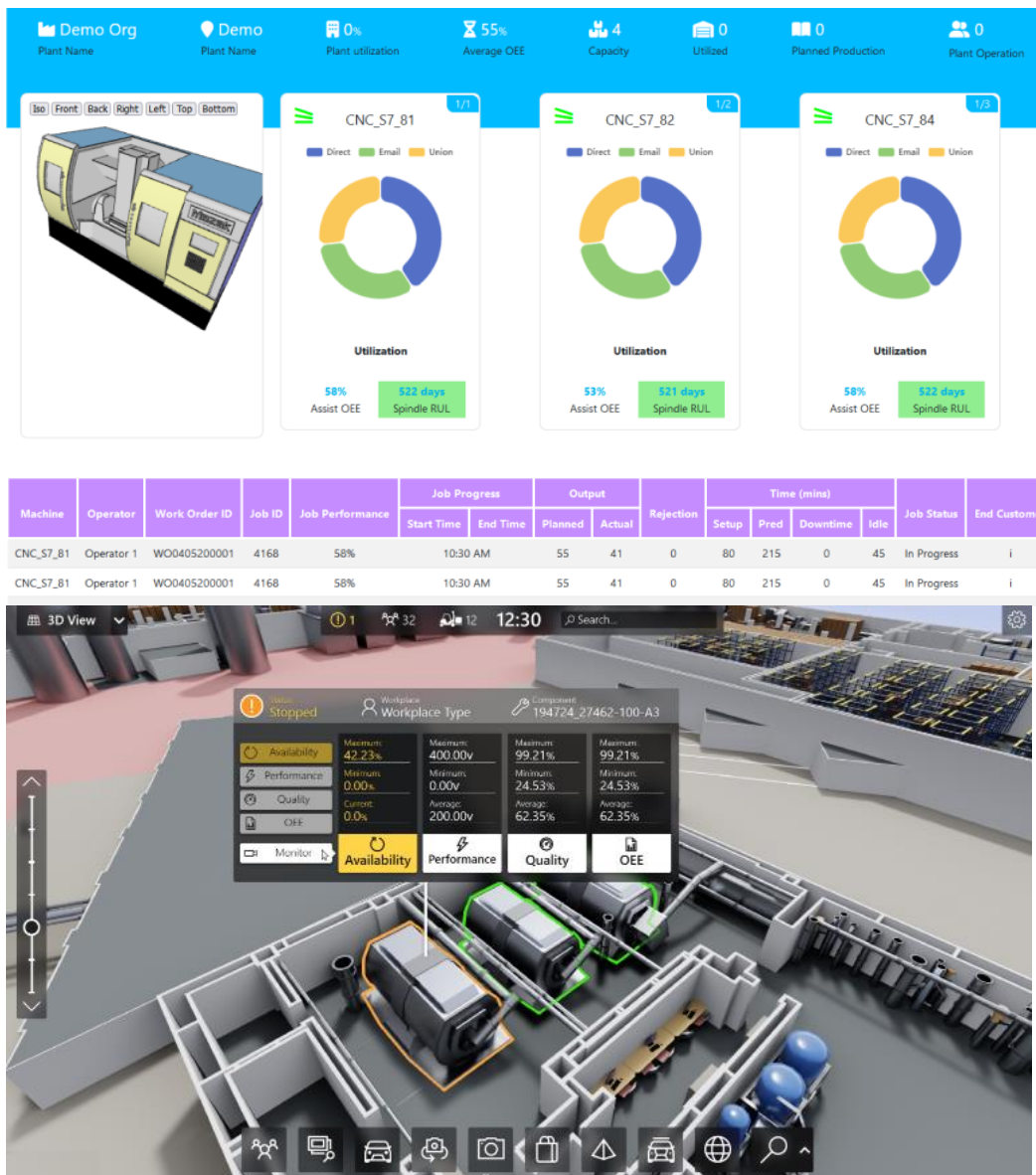
iii. **LoRaWAN™** based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv.   Predictive Maintenance

UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.

Fix when equipment is down.

Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.

Reactive

Preventive

Condition Monitoring

Predictive

Scheduled maintenance

Use analytics to predict failures.

2.2   About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.
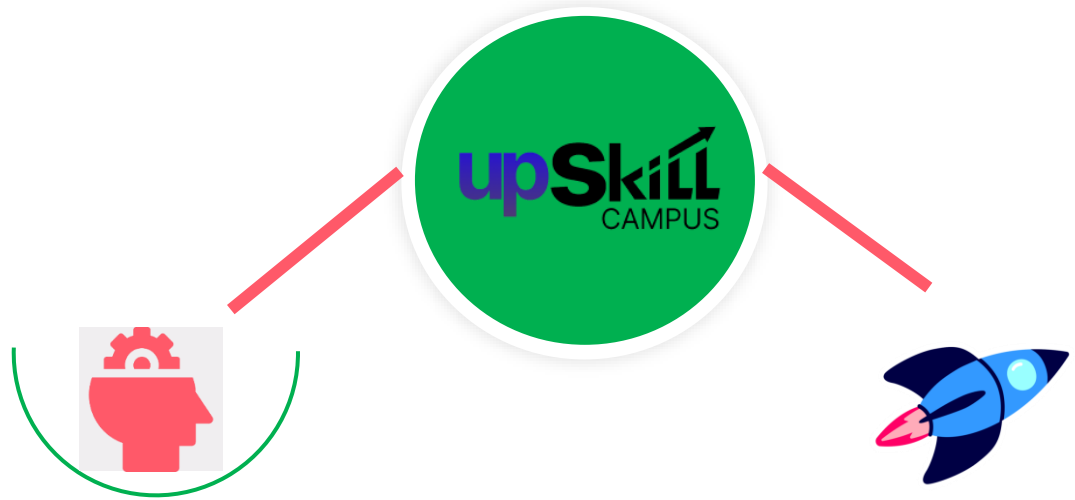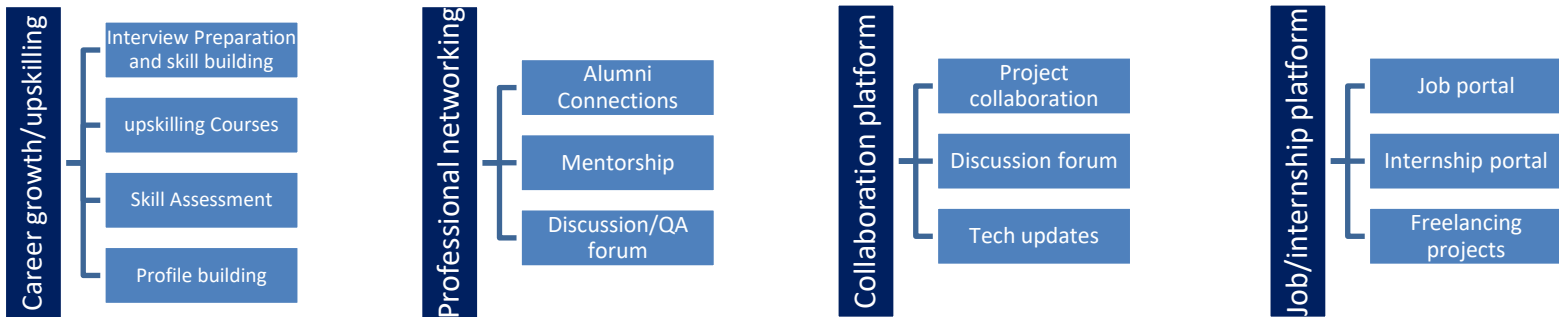
Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

https://www.upskillcampus.com/

Career growth/upskilling	Professional networking	Collaboration platform	Job/internship platform
Interview Preparation and skill building	Alumni Connections	Project collaboration	Job portal
upskilling Courses	Mentorship	Discussion forum	Internship portal
Skill Assessment	Discussion/QA forum	Tech updates	Freelancing projects
Profile building			

2.3   The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

☛ get practical experience of working in the industry.

☛ to solve real world problems.

☛ to have improved job prospects.

☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] The Java™M2 Tutorials

The Java tutorials are practical guides for programmers who want to use the Java programming language to create applications.

https://java.sun.com/docs/books/tutorial/index.html

[2] Sun Developer Network

Sun Microsystem's official website listing down all the API document

Technologies, books and other resources.

https://java.sun.com/reference/docs/

3 Problem Statement

Problem Statement for the Project: Music player application

To Develop a music player application that allows users to play, manage, and enjoy their music collection. The music player should provide essential functionalities for organizing and playing music files in various formats.

Required Functionalities:

1. Music File Import: Enable users to import music files from their local storage or specified directories into the music player library. The player should support common audio file formats, such as MP3, WAV, and FLAC.

2. Music Playback: Implement the ability to play, pause, resume, and stop music playback. The player should provide controls for adjusting the volume and seeking within the track.

3. Playlist Management: Allow users to create and manage playlists, enabling them to group their favorite songs or create custom collections. Users should be able to add and remove songs from playlists and organize the playlist order.

4. Music Library Organization: Provide features to organize and categorize music files within the player's library. Users should be able to create folders, assign tags or metadata to songs, and search for specific songs or artists.

5. Audio Equalizer: Implement an audio equalizer that allows users to adjust the sound output according to their preferences. The equalizer should provide pre-defined presets and allow users to customize the equalizer settings manually.

6. Shuffle and Repeat: Include options for shuffling the playlist order and repeating individual tracks or the entire playlist.

7. Crossfade: Implement a crossfade feature that smoothly transitions between songs, creating a seamless listening experience.

8. User Interface: Design an intuitive and user-friendly interface that provides easy navigation, displays album art and song information, and includes controls for playback, playlist management, and other functionalities.

9. Metadata Display: Retrieve and display metadata information, such as song title, artist, album, and duration, for each music file in the player's library.

10. File Format Compatibility: Ensure the music player supports a wide range of audio file formats to accommodate different user preferences and file types.

Note: This is a high-level overview of the required functionalities for a music player application. Additional features and details can be added based on specific requirements and project scope.

Minimum Features for a Music Player:

1. Music Playback: Users should be able to play, pause, and stop music playback.

- Output: The music player starts playing the selected song and displays basic playback controls such as play, pause, and stop.

2. Playlist Management: Users should be able to create and manage playlists.

- Output: Users can create a new playlist, add songs to the playlist, and view and modify existing playlists.

3. Music Library Organization: Users should be able to browse and select songs from their music library.

- Output: The music player displays a list of available songs in the library. Users can select a song to play.

4. Basic Navigation: Users should be able to navigate through the music library and playlists.

- Output: Users can browse through their music library, view songs by artist, album, or genre, and switch between different playlists.

5. Audio Control: Users should be able to adjust the volume of the music playback.

- Output: Users can increase or decrease the volume of the music player, and the output audio volume changes accordingly.

The expected output for these minimum features includes a functional music player interface with basic playback controls, the ability to create and manage playlists, the ability to browse and select songs, basic navigation features, and audio control capabilities. The player should provide a seamless experience for users to play and manage their music collection with ease.

4   Existing and Proposed solution

summary of existing solutions provided by others, what are their limitations?

4.1.1 About Spotify

You will need a **Spotify Premium** account to play music from [Spotify], but you can use SomeGeetha without a premium account (to play your Spotify playlists from other sources, for example)

SomeGeetha is using the official [Spotify Android Auth] library and [Retrofit] to access the [Spotify Web API], i.e., to obtain user library and playlists. In order to play music from Spotify, SomeGeetha uses the [librespot-java] library.

Feature overview

SomeGeetha Player is developed by me alone, so the project cannot be tested on many devices and scenarios; if you find an issue, open one here.

- The app will open on your library, categorized as Artists, Albums, Songs, and Playlists (available in the navigation drawer).
- It allows you to manage libraries for songs (add/remove from source library) and playlists (add/remove from playlist, create/remove playlists)
- It supports Android 'dark theme' (the screenshots above are done on a dark themed system).
- It is completely free (there are no ads, no limited version)
- It caches the library locally, so launching SomeGeetha requires virtually no data (only refreshing tokens and status of sources servers)

- **(TODO)** It has a 'data saving' mode that allows you to listen to music while consuming very low mobile data (by not loading album arts)
- The search feature allows you to search the local library instantly
- The "explore" mode allows you to search and browse sources for new music (for example search all Spotify and look for new releases)
- It has a layout that can adapt for tablet users (landscape layout)
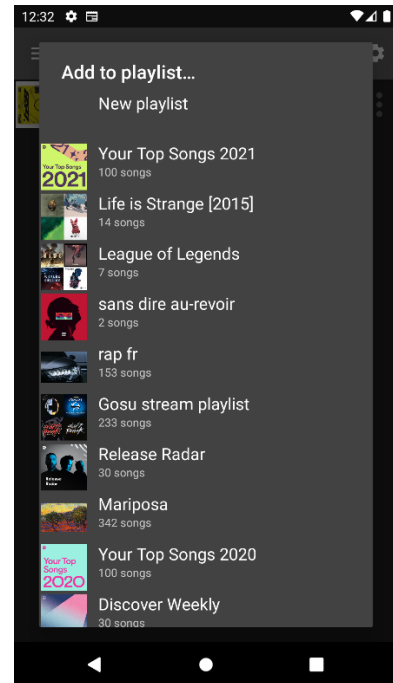
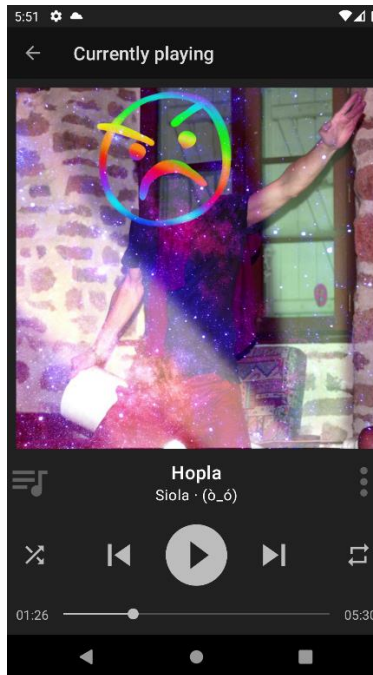It is available in different languages: English, French, German, Turkish
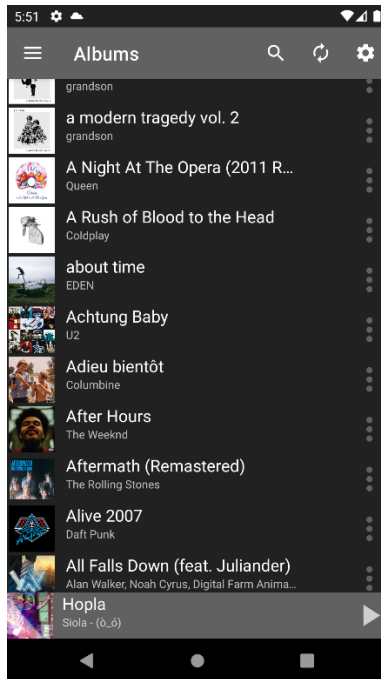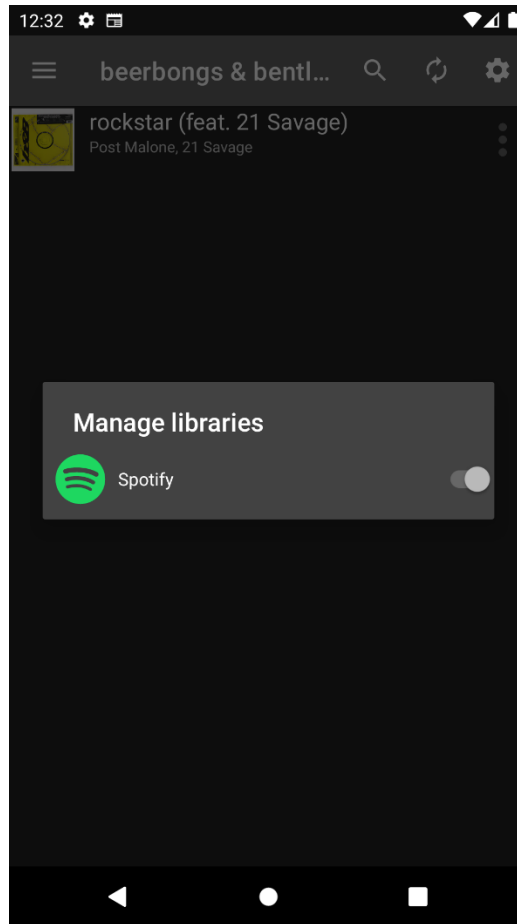
4.2 Code submission (GitHub link):
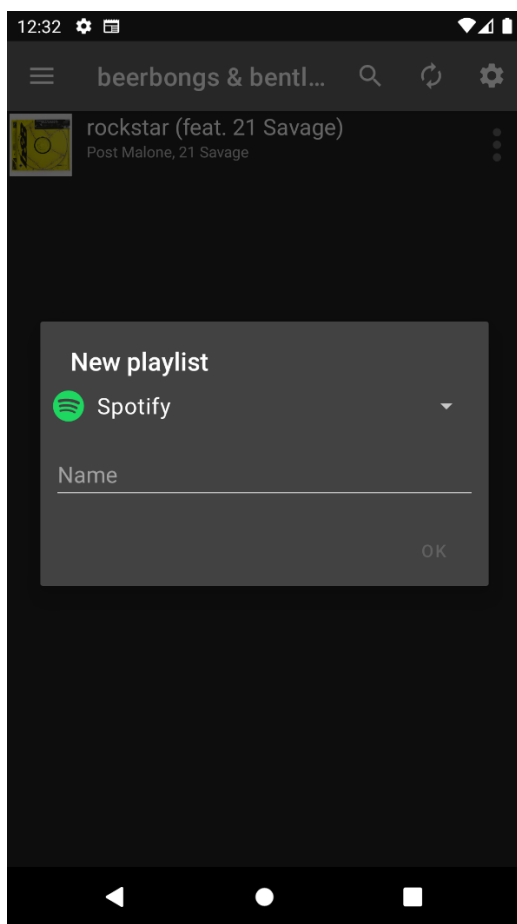https://github.com/Namithareddy105/SomeGeetha-music-player-application.git

4.3 Report submission (Github link) : first make placeholder, copy the link.

4.5 Interfaces

5 Performance Test

This is very important part and defines why this work is meant of Real industries, instead of being just academic project.

Here we need to first find the constraints.

How those constraints were taken care in your design?

What were test results around those constraints?

Constraints can be e.g., memory, MIPS (speed, operations per second), accuracy, durability, power consumption etc.

In case you could not test them, but still you should mention how identified constraints can impact your design, and what are recommendations to handle them.

5.1 Test Plan/ Test Cases

Introduction

The purpose of this report is to present the findings of the performance test conducted on the music player application. The performance test aimed to assess the application's responsiveness, resource utilization, and stability under various load conditions. This report provides an overview of the test objectives, methodology, results, and recommendations.

Test Objectives

The performance test for the music player application focused on the following objectives:

a. Measure the application's response time for basic operations like play, pause, skip, and seek.

b. Evaluate the application's resource utilization, including CPU, memory, and disk usage.

c. Assess the application's stability under different load scenarios.

d. Identify any performance bottlenecks or areas for improvement.

Test Environment

The performance test was conducted in the following environment:

Operating System: 22621.1992 ver: 22H2

Processor: Intel Core i5 12th Gen

RAM: 16.00 GB

Disk Space: SSD

Methodology

The performance test followed the following methodology:

a. Identified key user scenarios, including playlist creation, song search, playback controls, and metadata retrieval.

b. Developed test scripts to simulate user interactions and emulate realistic user behaviour.

c. Executed the test scripts on multiple test machines to generate load.

d. Monitored and recorded system resource utilization (CPU, memory, disk) during the test.

e. Captured response times for each test scenario and collected performance metrics.

f. Repeated the test under different load levels to assess scalability and stability.

g. Analysed the test results and identified any performance issues or bottlenecks

Test Results

The performance test results are summarized as follows:

a. Average response times for basic operations (play, pause, skip, seek) were measured and recorded.

b. Resource utilization metrics, including CPU, memory, and disk usage, were monitored during the test.

c. Stability was assessed by examining system behaviour under increasing load levels.

d. Performance bottlenecks and issues were identified, along with their impact on the application.

Recommendations

Based on the performance test results, the following recommendations are provided:

a. Optimize the response time for key operations to enhance user experience.

b. Monitor and optimize resource utilization to ensure efficient application performance.

c. Address any identified performance bottlenecks or issues to improve overall stability.

d. Conduct regular performance testing to proactively identify and resolve potential performance concerns.

e. Consider implementing caching mechanisms for frequently accessed data to improve response times.

Conclusion

The performance test conducted on the music player application provided valuable insights into its responsiveness, resource utilization, and stability under various load conditions. The test results highlighted areas of improvement and recommended actions to enhance the application's performance. By implementing the suggested recommendations, the music player application can deliver a smoother user experience and improved overall performance.

6 My learnings

Introduction:

During my internship, I had the opportunity to work on a music player application, which involved developing new features, optimizing existing functionality, and collaborating with a team of experienced developers. The internship provided me with invaluable practical experience and deepened my understanding of software development and the music industry.

Understanding the Requirements:

Before diving into the development process, I spent time thoroughly understanding the requirements of the music player application. This involved studying the project specifications, conducting research on existing music players, and identifying key features to be implemented. This initial stage taught me the importance of clear requirements gathering and effective communication with stakeholders.

Programming Languages and Technologies:

During the internship, I had the opportunity to work with java programming language and technologies. The primary languages used were JAVA, and I gained proficiency in implementing features using these languages. Additionally, I learned about libraries and frameworks commonly used in music player development. Understanding these technologies helped me enhance my programming skills and adapt to industry-standard practices.

Feature Development:

One of the key aspects of my internship was developing new features for the music player application. This involved working on tasks assigned by senior developers, implementing user interface components, integrating APIs for music streaming services, and enhancing the functionality of the application. By actively participating in feature development, I learned about modular coding, version control systems, and the importance of writing clean, maintainable code.

Bug Fixing and Testing:

Throughout the internship, I was involved in identifying and fixing bugs in the music player application. I learned about the importance of thorough testing and the different testing techniques, including unit testing and integration testing. Collaborating with the quality assurance team, I gained experience in debugging, troubleshooting, and ensuring the application's stability and performance.

Collaboration and Teamwork:

Working within a team of experienced developers and designers exposed me to effective collaboration and teamwork. I participated in daily stand-up meetings, contributed to discussions, and actively sought guidance from senior team members. This experience not only improved my technical skills but also taught me the significance of effective communication, task delegation, and adapting to different work styles.

Time Management and Prioritization:

Throughout the internship, I had the opportunity to work on multiple tasks simultaneously. This experience taught me the importance of effective time management and prioritization. I learned to balance competing priorities, break down complex tasks into manageable sub-tasks, and meet deadlines consistently. These skills will be invaluable in my future career as a software developer.

Conclusion:

My internship experience working on a music player application provided me with practical knowledge, technical skills, and exposure to real-world software development practices. I gained a deeper understanding of UI/UX design principles, programming languages, and technologies relevant to music player development. Furthermore, I developed essential skills in collaboration, time management, and bug fixing. This internship has been instrumental in shaping my career aspirations and has laid a strong foundation for my future growth as a software developer in the music industry

7   Future work scope

Personalized Recommendations: Enhance the music player's recommendation system by implementing machine learning algorithms and user profiling techniques. The application can analyze user preferences, listening history, and behavior patterns to provide personalized music recommendations and curated playlists.

Integration with Streaming Platforms: Integrate the music player with popular music streaming platforms like Spotify, Apple Music, and Tidal. This integration would allow users to access their favorite tracks, playlists, and albums from these platforms directly within the music player application, providing a seamless experience.

Social Sharing Features: Implement social sharing features to enable users to share their favorite songs, playlists, and music discoveries with friends and followers. Integration with social media platforms such as Facebook, Twitter, and Instagram can enhance the user's social music experience and help in discovering new music through their network.

Lyrics and Synchronized Playback: Incorporate lyrics feature that displays synchronized lyrics while playing songs. This feature can enhance the user's experience by enabling them to sing along or understand the song's meaning more effectively. Additionally, provide an option to contribute and edit lyrics to create a collaborative platform for music lovers.

Podcast and Audiobook Support: Extend the music player's capabilities to include support for podcasts and audiobooks. This expansion would allow users to seamlessly switch between music, podcasts, and audiobooks, providing a comprehensive audio entertainment experience within a single application.

Smart Speaker Integration: Integrate the music player application with popular smart speakers like Amazon Echo, Google Home, and Apple HomePod. This integration would enable users to control and play music using voice commands, providing a hands-free and convenient experience.

Enhanced User Interface and Customization: Continuously improve the user interface (UI) and user experience (UX) of the music player application. Consider introducing themes, customizable layouts, and gesture-based controls to provide a visually appealing and intuitive interface for users to interact with the app.

Integration with Music Events and Concerts: Collaborate with music event platforms and ticketing services to provide users with information about upcoming concerts, festivals, and live performances based on their music preferences. Enable ticket purchasing and integration with digital ticketing platforms for a seamless experience.

Remember, these are just some potential ideas for the future work scope of a music player application. The specific features and enhancements would depend on market trends, user feedback, and technological advancements in the coming years.

Presented by:

Name	College	Branch	Email id
1. G.Namitha reddy	NIEIT college, Mysore	ISE	namithareddyg95@gmail.com
2. Indu priya shirisha	JNNCE, Shimoga	CSE	indureddyg1997@gmail.com
3. N.C.Supritha	RYMEC, Ballari	ISE	ncsuprithancsupritha@gmail.com