

Industrial Internship Report on Music Player Application in Core Java

Prepared by
Pubayan Sanyal

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 to develop a music player application in core Java. The application contains basic playback controls for music like play, pause, resume, next song and previous song. It contains the functionality of making music folders and music playlists where users can custom design their folders or playlists with selected songs. The repeat track option exists for a song as well as a playlist. The user has to first import local music files into his/her music library to play the local music songs. Designing a functional UI in this project was a challenge I overcame.

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	4
2	Introduction	6
2.1	About UniConverge Technologies Pvt Ltd	6
2.2	About upskill Campus.....	10
2.3	Objective	12
2.4	Reference	12
2.5	Glossary.....	12
3	Problem Statement.....	13
4	Existing and Proposed solution	14
5	Proposed Design/ Model	15
5.1	High Level Diagram (if applicable)	15
5.2	Low Level Diagram (if applicable).....	17
5.3	Interfaces (if applicable).....	19
6	Performance Test	21
6.1	Test Plan/ Test Cases	21
6.2	Test Procedure.....	21
6.3	Performance Outcome.....	22
7	My learnings.....	23
8	Future work scope	24

1 Preface

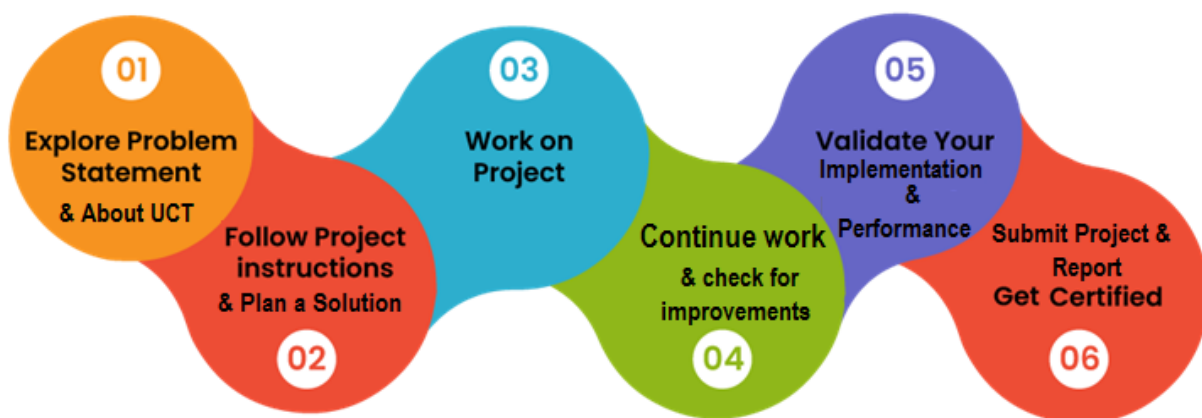
In the past 6 weeks, every week I learned something new from this internship. In the first week, I studied about UniConverge Technologies, on what domains they work and they address which challenges using which technologies. In the second week, I prepared a UI design for my project using figma. In the third week, I mainly studied about java packages and library code that I could use to implement the music functionalities in my project. The fourth, fifth and sixth weeks I devoted to trying out the code from different java libraries, testing their output and which of them best gives the output meeting the criteria of my project.

An industry relevant internship is very important in my career development as it would give me an exposure to the industry standards which I must meet in the coming future. It would require me to upskill myself in relevant technology domains, thus sharpening my skills in fields like writing good and clean code of industry accepted programming language, which in this case is java.

My project statement is to build a music player application in core java. The user must be able to import local music files into the music library and must be provided with basic playback controls like play, pause, resume, stop, next song and previous song. The application must have the feature of making custom songs music folders and music playlists and the option to repeat a single track or a playlist must be there. The app should provide a friendly user interface for the user to do the above.

UCT has provided me the opportunity to build the above mentioned project, thus exposing me to an industry relevant internship. I am grateful for this internship opportunity to UCT.

How Program was planned – this is depicted in the diagram below.



I learned a lot from this internship. I learned about the level of an industry relevant project. I learned about various java concepts and about various java packages and java libraries which exist in the market to make music playing functional in a java powered application. I learned about the various types of

outputs from different java music library codes and how to integrate library code into my project using jar files. Overall it was a great and enriching learning experience.

I am thankful to my friends and family who supported me all along while building this project.

My message to my juniors and peers would be to keep upskilling yourself and keep learning and taking part in internship programs.

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.



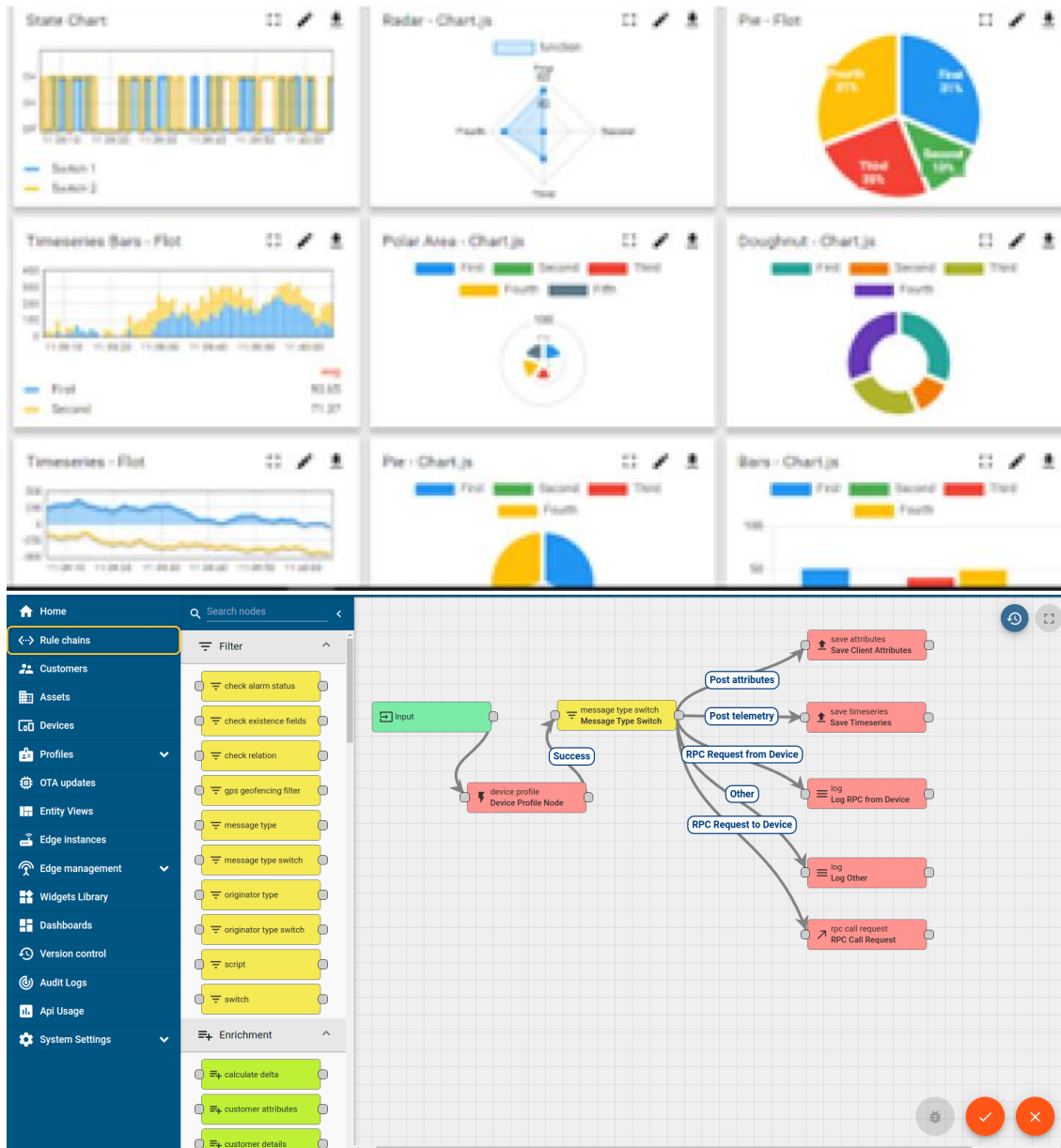
i. UCT IoT Platform ()

UCT Insight 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



FACTORY WATCH

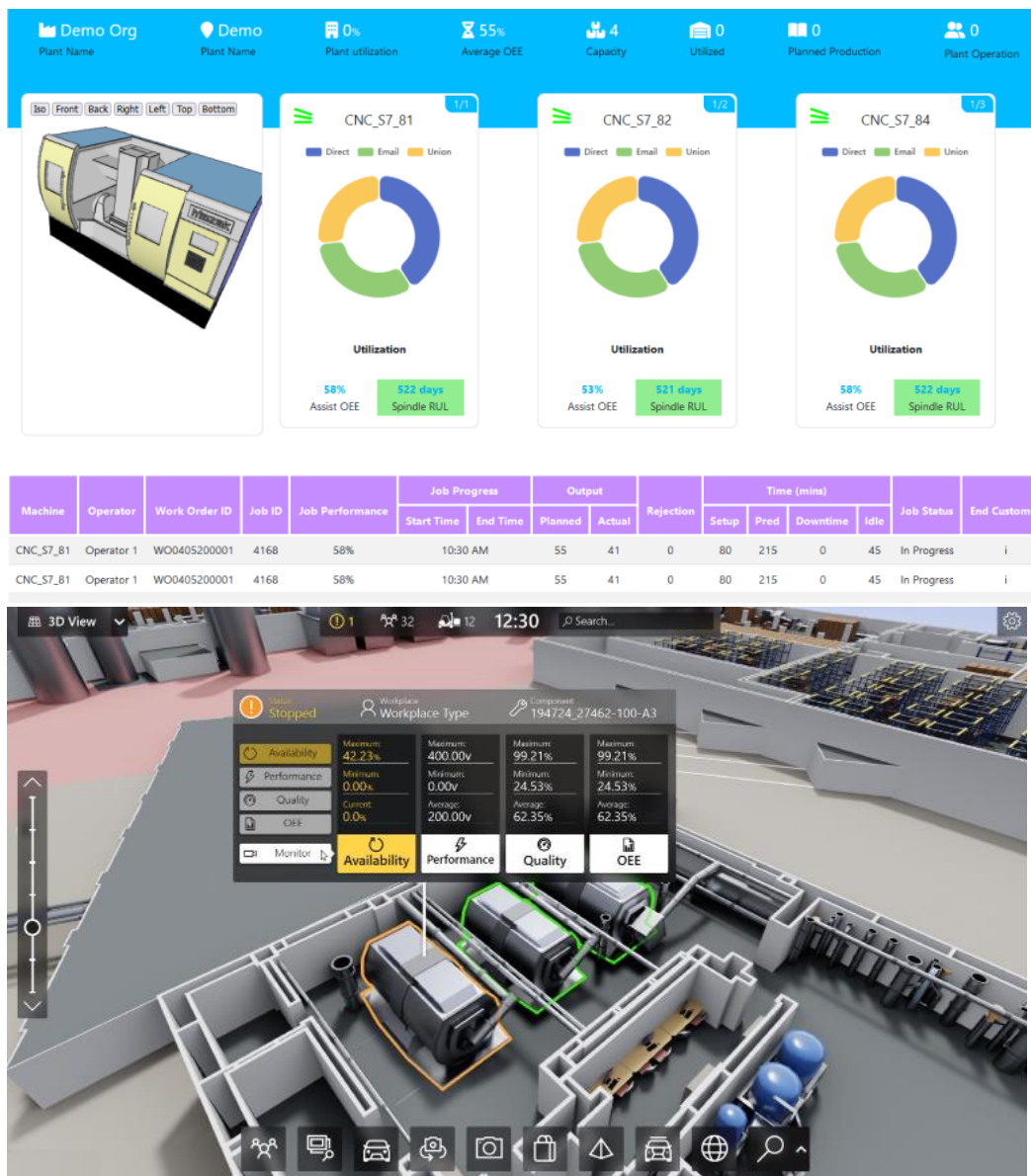
ii. Smart Factory Platform ()

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 unleash 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 want 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 technology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

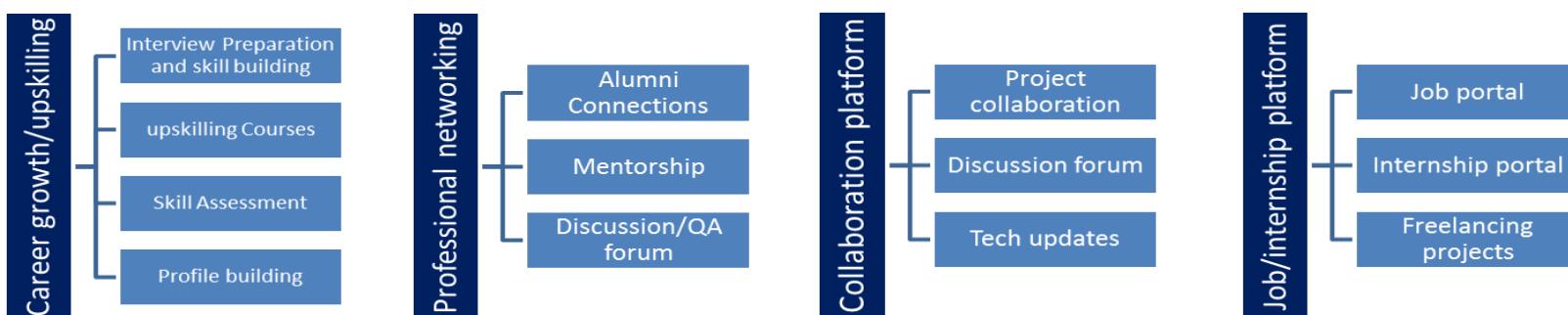
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.



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] Java learning tutorials like javatpoint.com and gfg (geeks for geeks)
- [2] Youtube channels on java app development

2.6 Glossary

Terms	Acronym
UCT	Uniconverge Technologies
GUI	Graphical User Interface
Java AWT	Java Abstract Window Toolkit

3 Problem Statement

In the assigned problem statement ,

I had to design a music player application in core java. I will be listing the essential features below :-

1. The user has to be able to import local music files from his/her device to the music library of the music player application in order to be able to play the music file.
2. The music library should list the songs along with added metadata of the songs like name of the song, name of the artist, name of the album and duration of the music file.
3. There should be basic playback controls provided to the user like play music, pause, resume, stop, next song and previous song. There should be a slider for the user to navigate through a song.
4. The user should be able to make custom folders as well as custom playlists and name them and then select songs to be included in those folders or playlists as per their wish. The option to repeat the entire playlist or to repeat a single track should be there.
5. There should be a user friendly interface for the user to be able to easily navigate and avail the above mentioned functionalities.

4 Existing and Proposed solution

There do exist a few solutions over the web. Different solutions use different java packages. Some have used java AWT and swing to develop their UI and have used library classes from the java media framework (JMF) package (javax.media) to get their music functionality. Similar to javax.media, others have used javafx and other java music functionality developed packages to implement music play. Some solutions are just to play a single selected music file without any importing or playlists functionality while some solutions are to play music available from the APIs of existing large music companies in the market, whereas others are to play local music files whose locations are included in the source program.

My proposed solution is not to use APIs of existing music companies but to give the user a feature in the application interface to import local music files to be played in the application, thus removing the requirement to hard code the location of local music files in the source code file, enhancing more versatility. My design has the feature of making music folders as well as music playlists after selecting custom songs as opposed to just playing a single music file, giving more enhancing features to the user. The user, however will have to download local music files to be played using the application.

Value addition would be to give the user an immersive experience for his/her local music files stored in his/her computer using the features of my music player application.

4.1 Code submission (Github link)

Below I am submitting my github repository link where I have upload my source code files :-

<https://github.com/Pubayan-Sanyal/Java-Internship.git>

4.2 Report submission (Github link) :

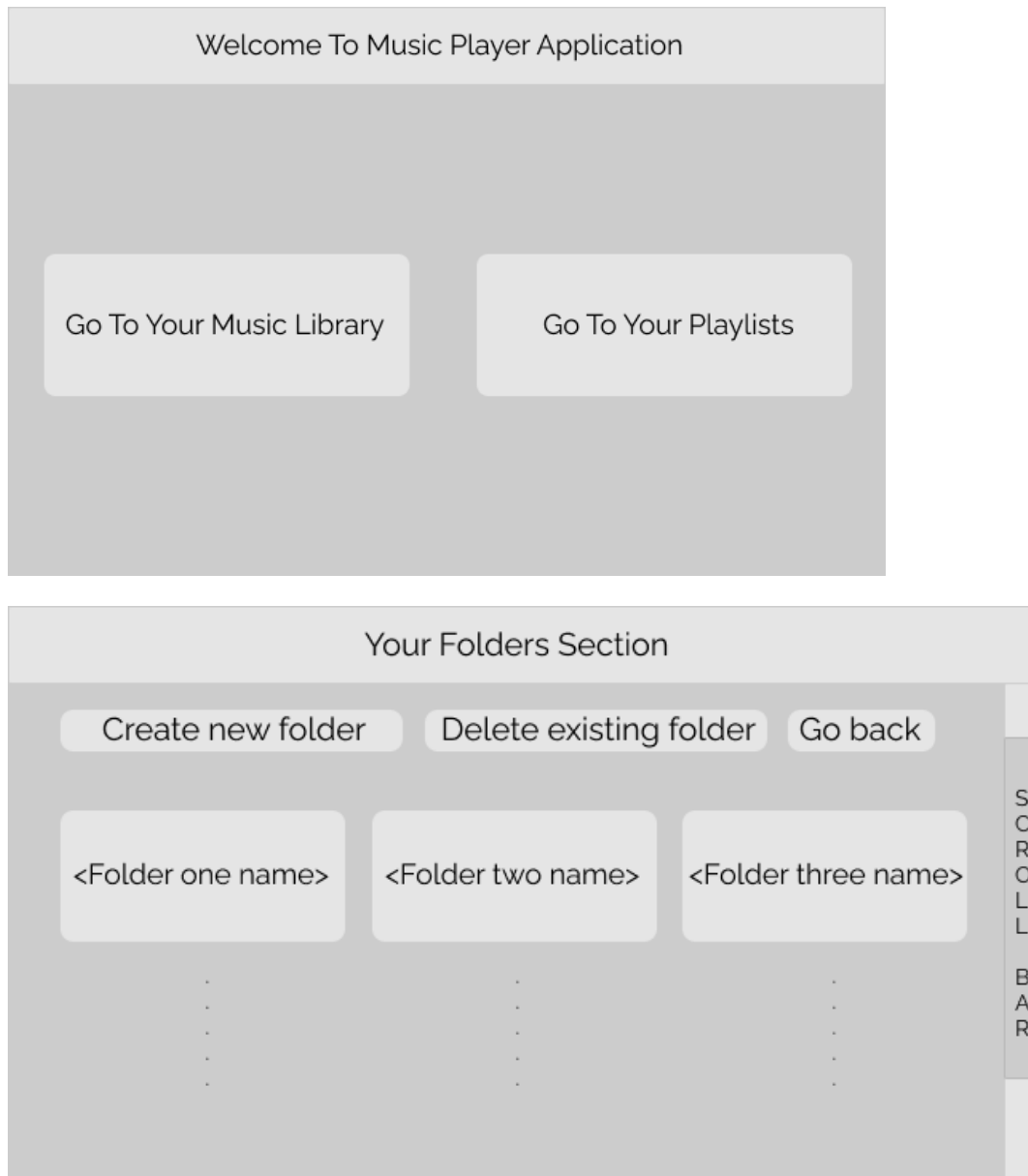
Below I am submitting my github repository link where I have uploaded this internship report itself :-

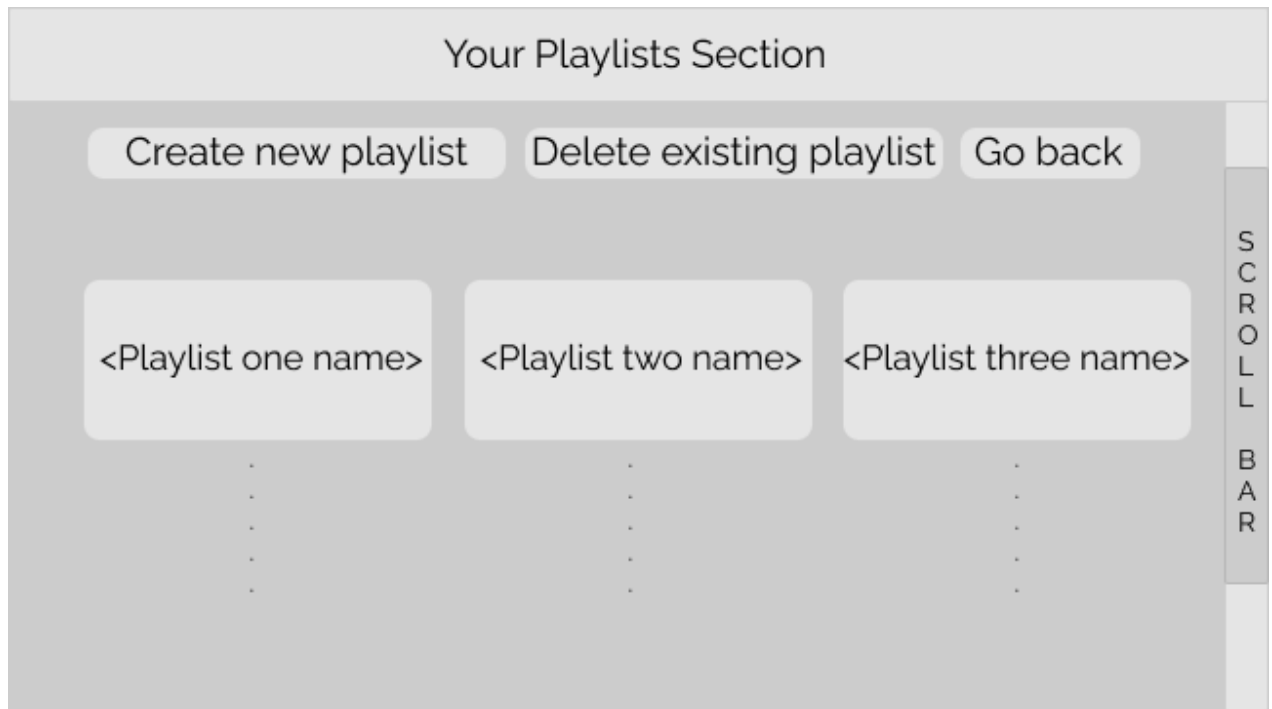
<https://github.com/Pubayan-Sanyal/Core-Java-Internship-Report.git>

5 Proposed Design/ Model

Below I am providing diagrams of my proposed design, which would include the details of my design, inherent from the diagrams.

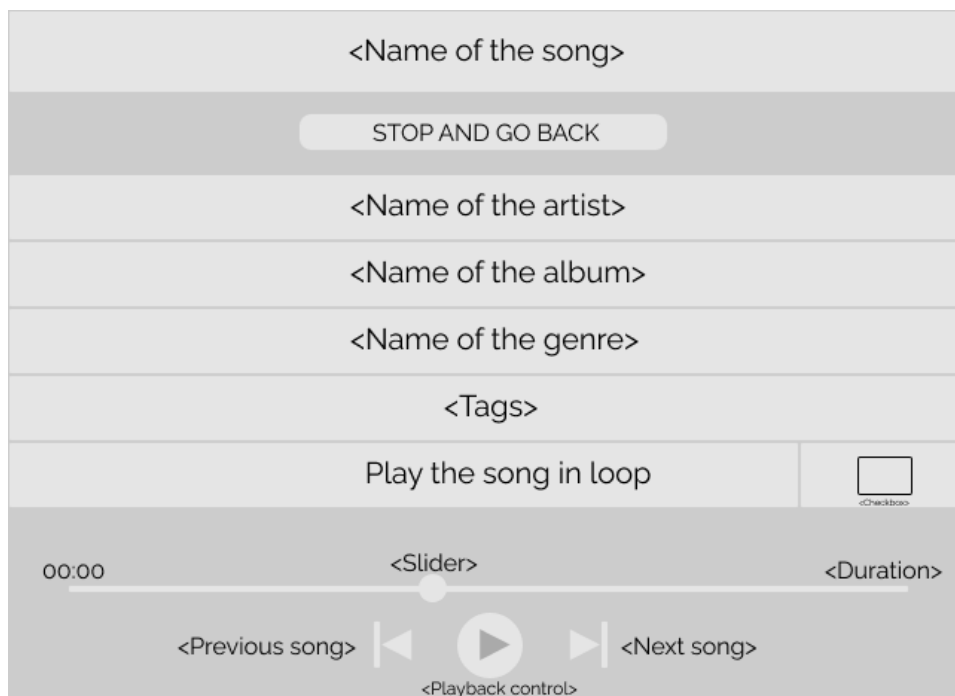
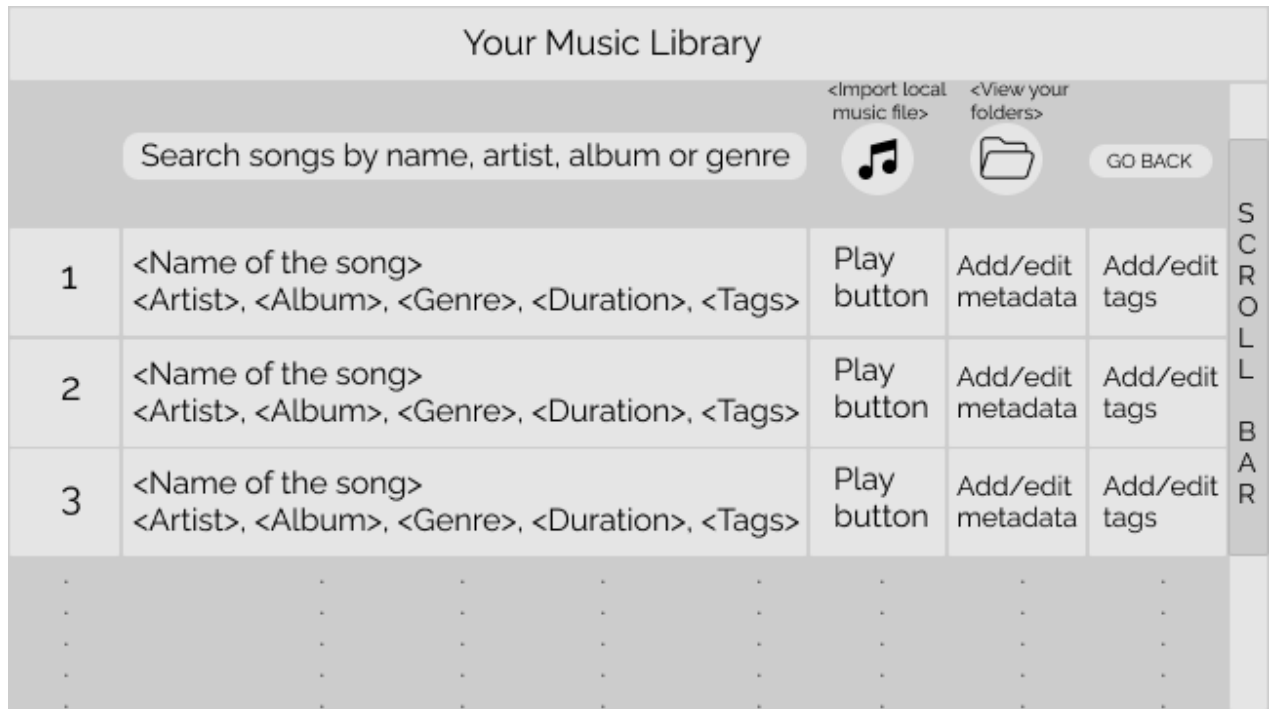
5.1 High Level Diagram (if applicable)





Above are high level diagrams of the system

5.2 Low Level Diagram (if applicable)



Name of the folder				
<div>Add song</div> <div>Remove song</div>				
<div>Change folder name</div> <div>Go back</div>				
<div>Search songs by name, artist, album or genre</div>				
1	<Name of the song> <Artist>, <Album>, <Genre>, <Duration>, <Tags>	Play button	Add/edit metadata	Add/edit tags
2	<Name of the song> <Artist>, <Album>, <Genre>, <Duration>, <Tags>	Play button	Add/edit metadata	Add/edit tags
.
.
.
.
.

SCROLL BAR

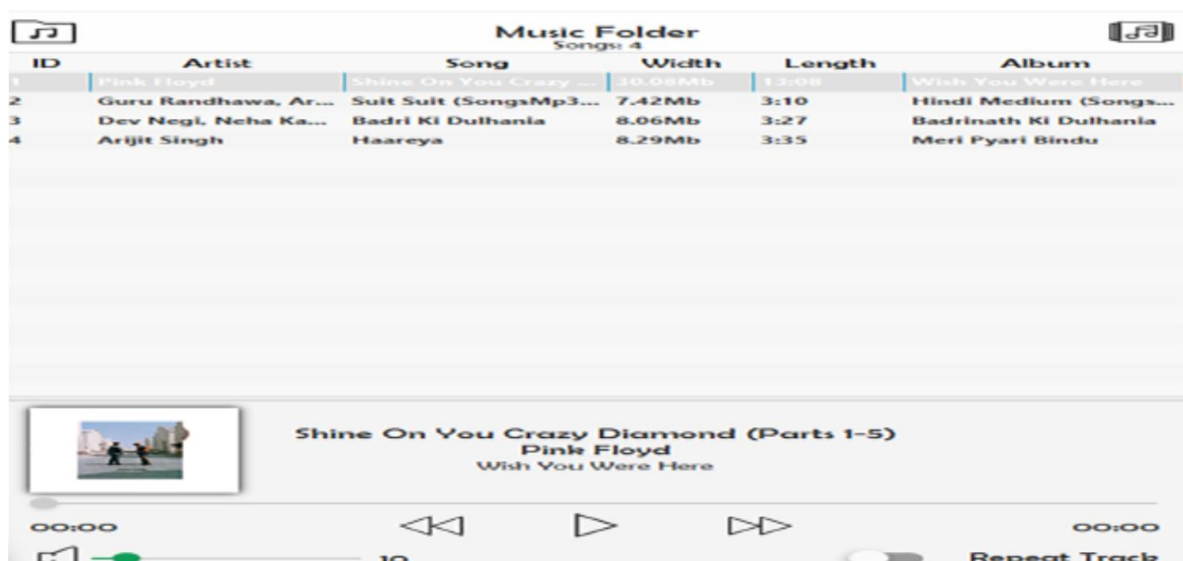
Name of the playlist				
<div>Add song</div> <div>Remove song</div>				
<div>Change playlist name</div> <div>Go back</div>				
<div>Search songs by name, artist, album or genre</div>				
<div>Change the order of songs in the playlist</div>				
<div>Click here to start playing the playlist</div>				
Play the playlist in loop		<input type="checkbox"/> <checkbox>		
Shuffle playlist while playing		<input type="checkbox"/> <checkbox>		
1	<Name of the song> <Artist>, <Album>, <Genre>, <Duration>, <Tags>	Play button	Add/edit metadata	Add/edit tags
2	<Name of the song> <Artist>, <Album>, <Genre>, <Duration>, <Tags>	Play button	Add/edit metadata	Add/edit tags
.
.
.
.

SCROLL BAR

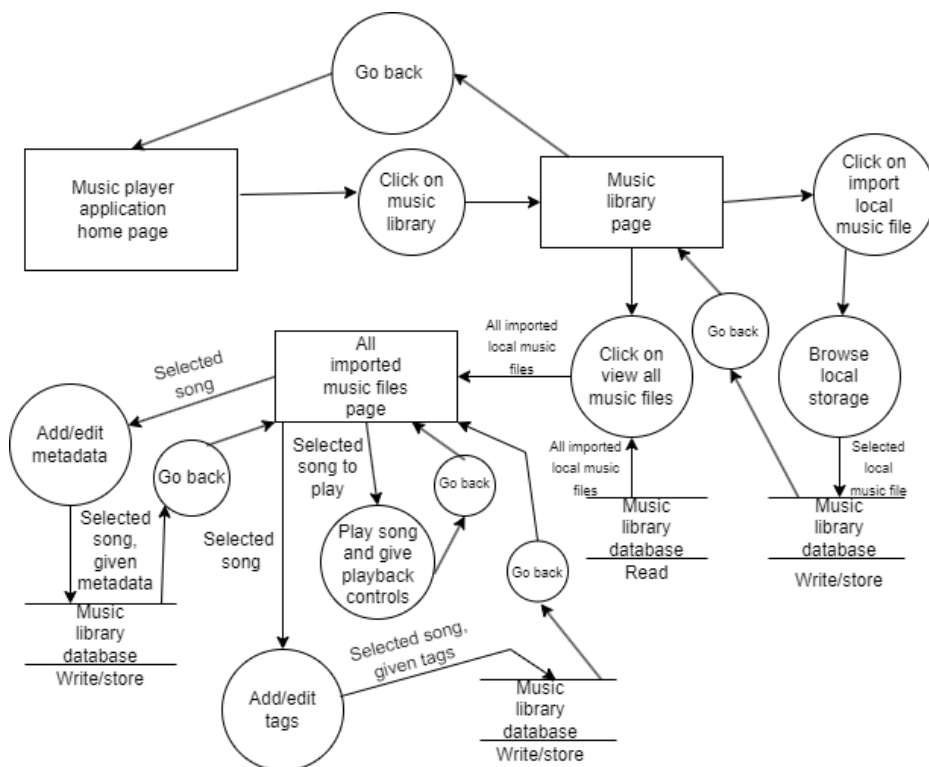
Above are low level diagrams of the system

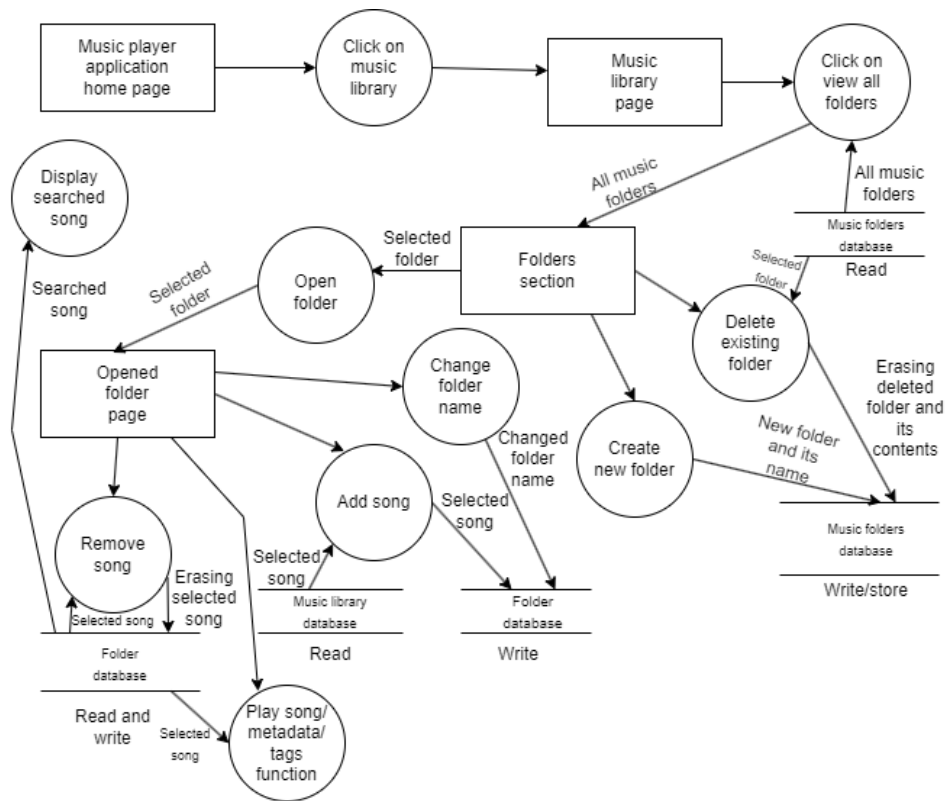
5.3 Interfaces (if applicable)

Interfaces are already evident from the high level and low level diagrams of the system I provided. I am also providing the image of the functional UI from my source code below.



Below I am providing data flow diagrams for my system :-





6 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.

Constraints include device memory taken by the application, the speed at which the application responds, accuracy and durability of the application and power consumption of the application.

Test results :-

Standard device storage (like 512GB) or more is more than sufficient. The application itself shall not require more space than 500MB space (space for source code files), along with that it will take the memory space required for the imported local music files. Many music files still should be accommodated within standard local device storage. RAM of 8GB should be sufficient for proper running of the music player, though 4GB RAM will also do.

The application shall respond as soon as we click on any feature, making it speed efficient. The application shall not buffer with 8GB RAM or more.

Proper working features do ensure the accuracy of the application and since it runs on java source code files, no change in the java source code files ensures the durability of the application over time.

It is not a high power consumption application as it is a simple music player and the same can be ensured through operating system checking credentials present in the computer on which the application runs.

6.1 Test Plan/ Test Cases

Test plan/test cases would be to check the functionalities of the music player application while keeping the above mentioned constraints in mind. The requirements to be tested are the ability to import local music files into the application, whether the import feature is causing any bugs while file searching or not. The playback controls are to be tested whether play, pause, resume and so on cause any technical issues. Making folders and playlists and repeating tracks smoothly are remaining tests. Operating system resources usage tracing can be done in the computer system in which the application runs to test the remaining constraints which are as mentioned above.

6.2 Test Procedure

Functionalities of the application can be tested by clicking on the buttons or icons which enable the features in the application and then tracing the working of the functionalities, as to whether they work with any bugs, delays or runtime difficulties. Application storage can be checked in the file manager of the host system to see the memory benchmarks of the application, as to what is the

total space occupied by the exe file along with cache storage for imported files. Features like speed can be tested by tracking time taken for the application to open and tracking time for each feature to work. Features like RAM usage can be traced with the host system checking features available to test the RAM usage for any application. Host OS system checking functions can also be used to check the power consumption. Consistent source file storage should ensure accuracy and durability of the application.

6.3 Performance Outcome

Test performance outcome is that all features do work properly for the application. Sometimes there could be a few execution time errors while running the java source files which need to be removed. Possible errors could pop up during the run time of the java source files using different IDEs and operating systems (like execution command errors) which have to be resolved. As of memory, standard local device storage which comes integrated with host computer will do the work. RAM of 8GB or more is sufficient for the application and power consumption is not very high. Speed of the application is standard depending on RAM usage availability and no changes in source code files ensure the accuracy and durability of the application.

7 My learnings

My learnings include :-

1. Understanding the standards and the criteria of an industry level project, including constraint complexity, how to research to complete a project and checking its performance.
2. Learning OOPs concepts and java concepts in depth which are essentially asked in job interviews and are required to build applications.
3. Learning about various java GUI tools like the java AWT , java swing and javafx, comparison of the GUIs created through them and their implementation and advantages. My learnings include which classes are to be used from java GUI packages like AWT, Swing and javafx and which functions are to be called to implement a functional GUI.
4. Applying various java frameworks to play music files in the java application created in the system, like java media framework (javax.media package) and javafx music playing tools and which media playing package better suits the needs of my project. Learning to play music files using java classes was completely new to me.
5. Learning to add library files like java JAR files to my java project and integration of the library files and java JAR files into my project, including the addition of javafx runtime components to deal with execution command errors in the IDE terminal.

8 Future work scope

Future work scope in this project can include enhancing the features of my music player application, making it like the music player application of large music companies. I would like to implement java database (JDBC) in the future for more efficient music file storage after learning JDBC. I would also like to use music APIs available over the web instead of cumbersome asking the user to download music files and import them locally. APIs would make user experience much more smoother. I would also like to enhance the functioning of my features which are available in my music app, making the features work with smoother transitions providing better user experience. Playback controls, folder making and playlist making can be enhanced using specific library classes available in developing java packages, which I would explore in the future.