

Industrial Internship Report on

“ Music player with playlist ”

Prepared by

[Amit Lanjewar]

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.

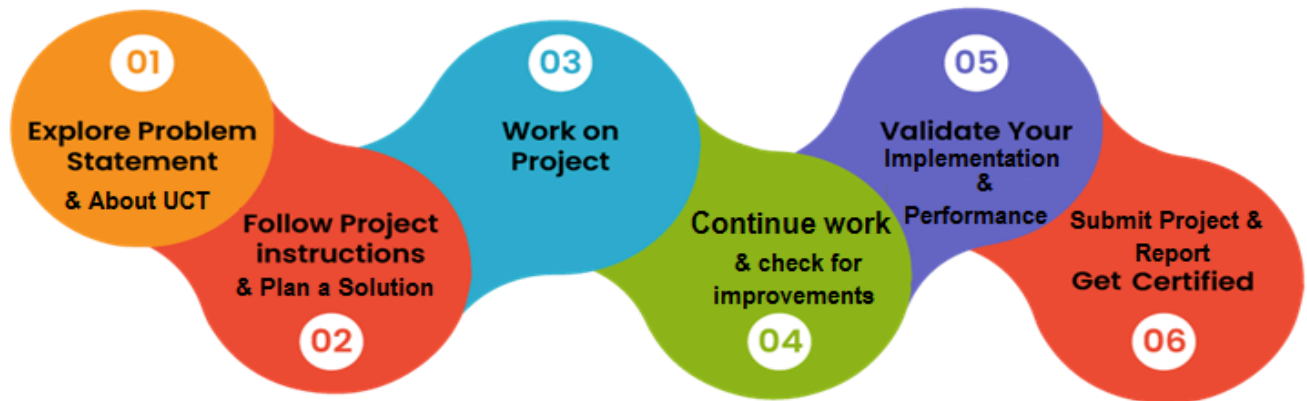
My project was (Tell about ur Project)

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.....	3
2	Introduction.....	4
2.1	About UniConverge Technologies Pvt Ltd.....	4
2.2	About upskill Campus.....	8
2.3	Objective.....	9
2.4	Reference.....	9
2.5	Glossary.....	10
3	Problem Statement.....	11
4	Existing and Proposed solution.....	12
5	Proposed Design/ Model.....	13
5.1	High Level Diagram (if applicable).....	13
5.2	Low Level Diagram (if applicable).....	13
5.3	Interfaces (if applicable).....	13
6	Performance Test.....	14
6.1	Test Plan/ Test Cases.....	14
6.2	Test Procedure.....	14
6.3	Performance Outcome.....	14
7	My learnings.....	15
8	Future work scope.....	16

Preface



Summary of the 6 Weeks' Work

Over 6 weeks, I gained hands-on experience with Java development, covering core concepts like object-oriented programming, interfaces, and database integration using JDBC. I worked on a **Music Player Application** that interacts with a MySQL database, applying what I learned through weekly tasks and quizzes.

Need of Relevant Internship in Career Development

This internship provided valuable practical experience, bridging the gap between theory and real-world application. It helped me improve my coding skills and better understand how Java is used in professional software development.

Project/Problem Statement

The project involved developing a **Music Player Application** that manages playlists and interacts with a MySQL database. This gave me practical experience with Java's core features and database handling.

Opportunity Given by USC/UCT

The **USC/UCT Internship** provided me with an excellent platform to learn and apply Java in a professional setting. It helped me build my technical skills and provided insight into the real-world use of programming languages.

Program Planning

The internship was well-structured, starting with Java fundamentals and progressing to more advanced topics like JDBC and object-oriented principles. Weekly reports and quizzes ensured I stayed on track.

Your Learnings and Experience

I gained a strong understanding of Java's core principles, especially in database management and object-oriented programming. Working on the music player project solidified these skills and enhanced my problem-solving abilities.

Thanks to All Who Helped You

Thanks to my mentor and my peers for their constant support and guidance throughout this internship.

1 Introduction

1.1 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.



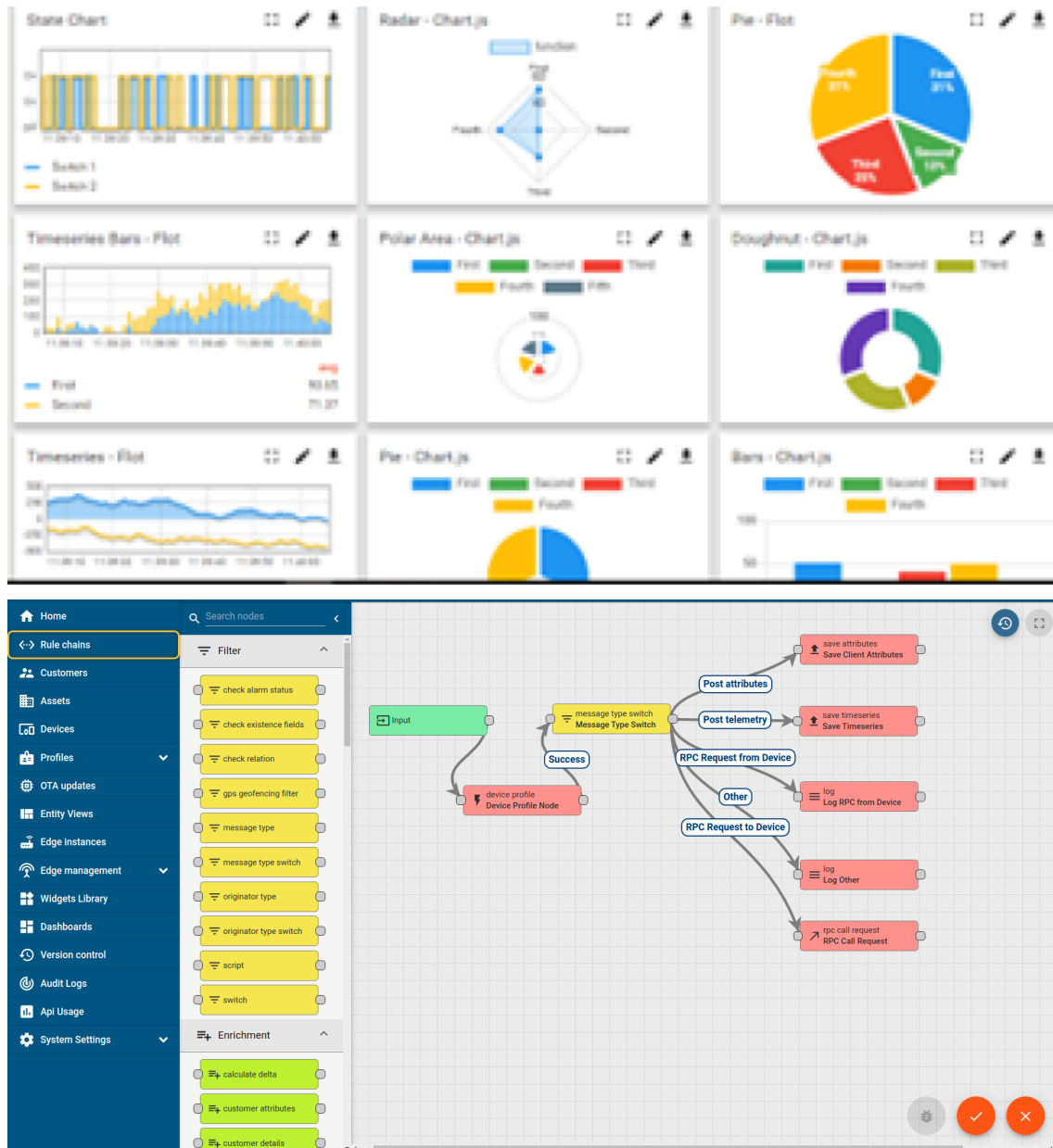
i. UCT IoT Platform (**Insight**)

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



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.



Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i



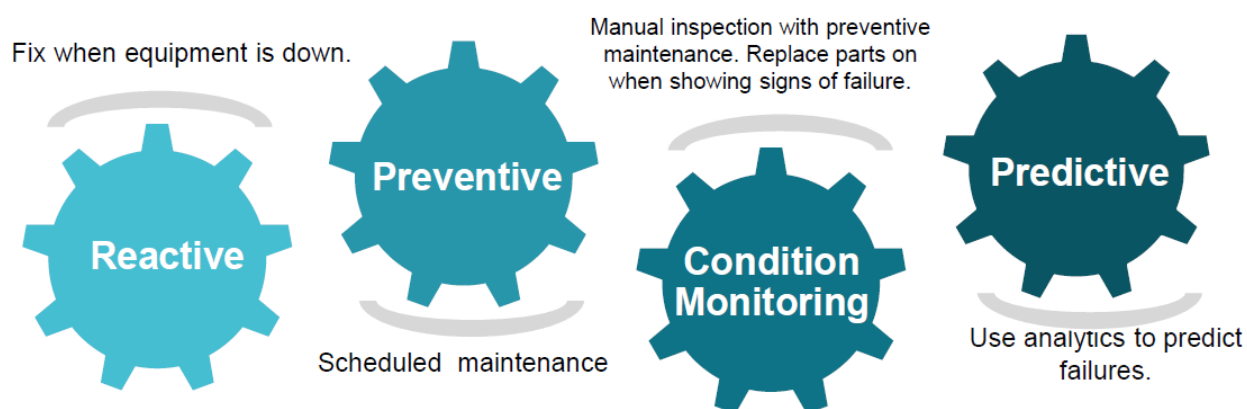


iii. LoRaWAN based Solution

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.



1.2 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.

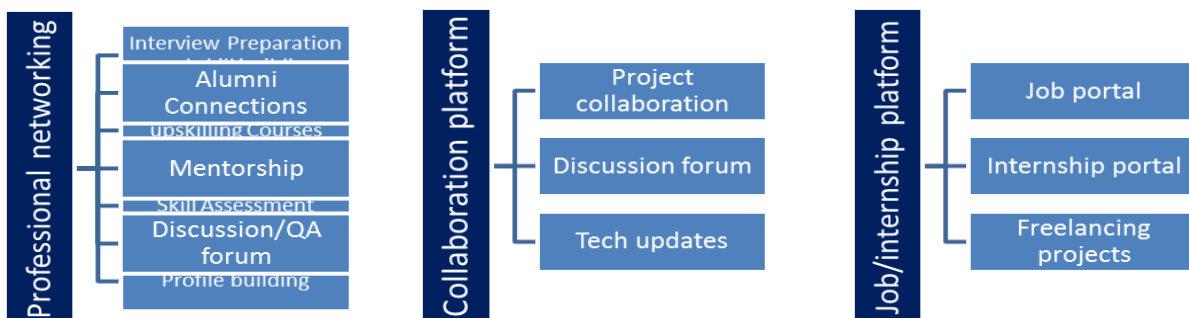


Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services



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

<https://www.upskillcampus.com>



1.3 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.

1.4 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.

1.5 Reference

- [1] Java Documentation - <https://docs.oracle.com/javase/tutorial/>
- [2] JDBC API Guide - <https://docs.oracle.com/javase/tutorial/jdbc/>
- [3] GitHub Repository - <https://github.com/yourusername/upskillcampus/>

1.6 Glossary

Terms	Acronym
JDBC (Java Database Connectivity)	- API (Application Programming Interface)
- SQL (Structured Query Language)	GUI (Graphical User Interface)
- MP3 (Audio File Format)	UI (User Interface)
Playlist	MVC (Model-View-Controller)
Looping	FIFO (First In, First Out)

2 Problem Statement

Problem Statement (Summary)

The project involves developing a **Music Player App** that allows users to manage their music collection and create playlists using **JDBC API**. The app should support features like music playback (play, pause, stop, skip), playlist creation, and storage of playlist data in a database for easy retrieval. Key functionalities include:

1. **Music Playback:** Play, pause, stop, and skip tracks.
2. **Music Library Management:** Store and retrieve song metadata (artist, album, duration) using JDBC.
3. **Playlist Management:** Create, modify, and delete playlists, with the ability to add, remove, and reorder songs.
4. **Search and Filtering:** Search by artist, genre, or custom tags using SQL queries.
5. **Error Handling:** Manage exceptions and database issues.
6. **User Interface:** Design a user-friendly interface for easy navigation.

The project focuses on building a fully functional music player with advanced playlist and database management.

3 Existing and

Proposed solution

Existing Solutions:

Many existing music player apps offer basic playback and playlist management. However, they often lack integration with databases, limiting the ability to store and manage metadata or playlists persistently. Additionally, advanced features like search, filtering, and error handling are often missing or not fully optimized.

Proposed Solution:

Our proposed solution is a **Java-based music player** that uses **JDBC API** for seamless integration with a database, allowing efficient storage and retrieval of song and playlist data. The application will offer features like advanced search, playlist creation, and management, along with robust error handling and an intuitive user interface.

Value Addition:

- **Database Integration:** Using JDBC for persistent storage of music data and playlists.
- **Advanced Search:** Filtering songs by artist, genre, and more.
- **Error Handling:** Robust mechanisms to handle exceptions and database issues.
- **User-Friendly UI:** Simplified interface for easy interaction.

3.1 Code submission (Github link)

<https://github.com/githubamit999/upskillcampus.git>

3.2 Report submission (Github link) :

[https://github.com/githubamit999/upskillcampus/blob/782d3e7b9eda369cc547937af6e84088c2b09b58/MusicPlayer Amit Lanjewar USC UCT.pdf](https://github.com/githubamit999/upskillcampus/blob/782d3e7b9eda369cc547937af6e84088c2b09b58/MusicPlayer%20Amit%20Lanjewar%20USC%20UCT.pdf)

4 Proposed Design/

Model

Proposed Design/Model

The music player application is designed to offer seamless music playback, playlist creation, and library management, with the following flow:

1. **Start:**
 - The application starts with a user-friendly interface showing options to play music, create playlists, and manage the music library.
2. **Intermediate Stages:**
 - **Music Playback:** Users can play, pause, skip, and stop tracks.
 - **Music Library Management:** Retrieves song data from the database using JDBC.
 - **Playlist Management:** Users can create, modify, and delete playlists, with data stored in the database.
 - **Search and Filtering:** Allows users to search for songs based on tags or metadata.
3. **Final Outcome:**
 - **Playlist Playback:** Users can play songs from any selected playlist.
 - **Database Persistence:** Changes to playlists are saved in the database.
 - **Error Handling:** Logs errors and informs the user of issues.

Technologies:

- **Java** for application logic.
- **JDBC** for database interaction.
- **MySQL** for database storage.
- **GUI** for user interaction.

Value Addition: The system ensures data persistence with MySQL, allowing users to manage their music and playlists across sessions.

4.1 High Level Diagram (if

applicable)

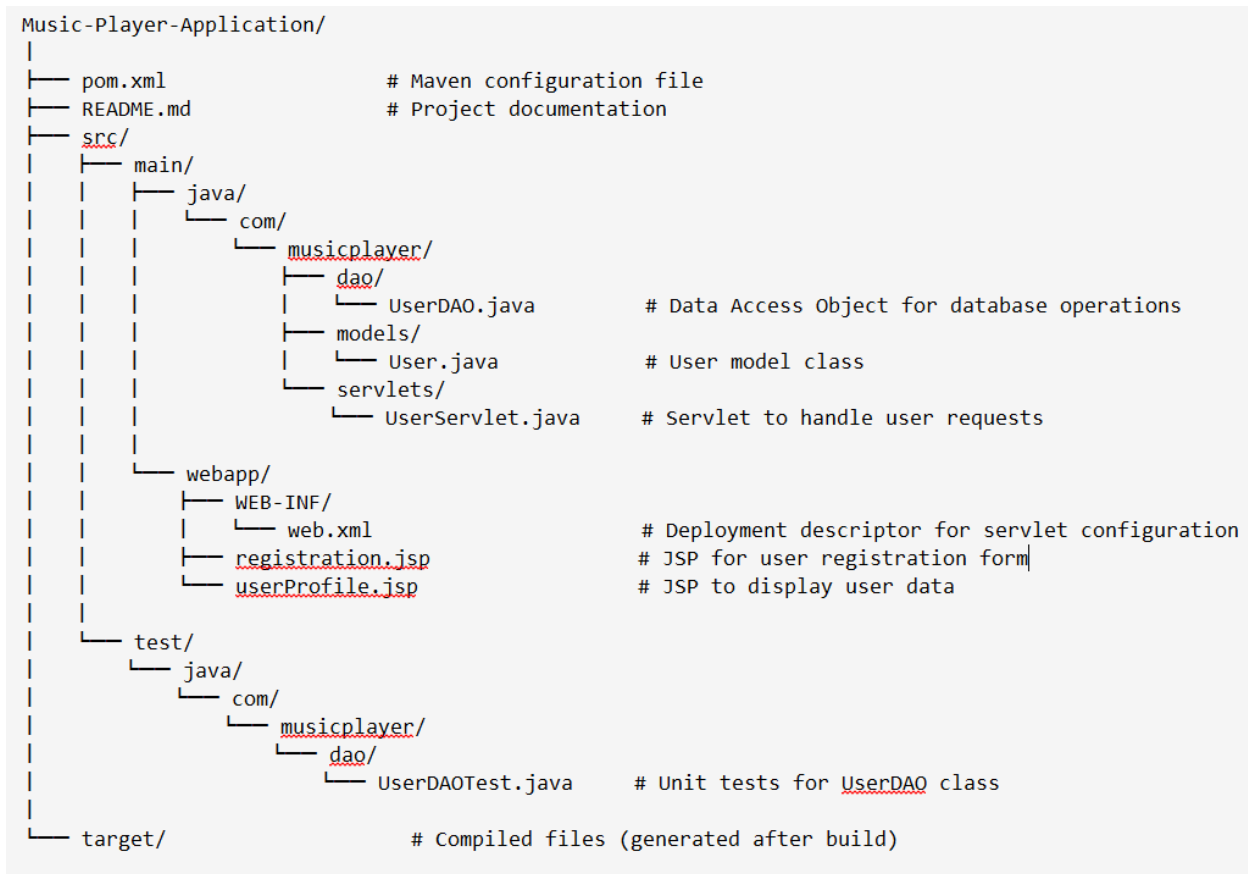
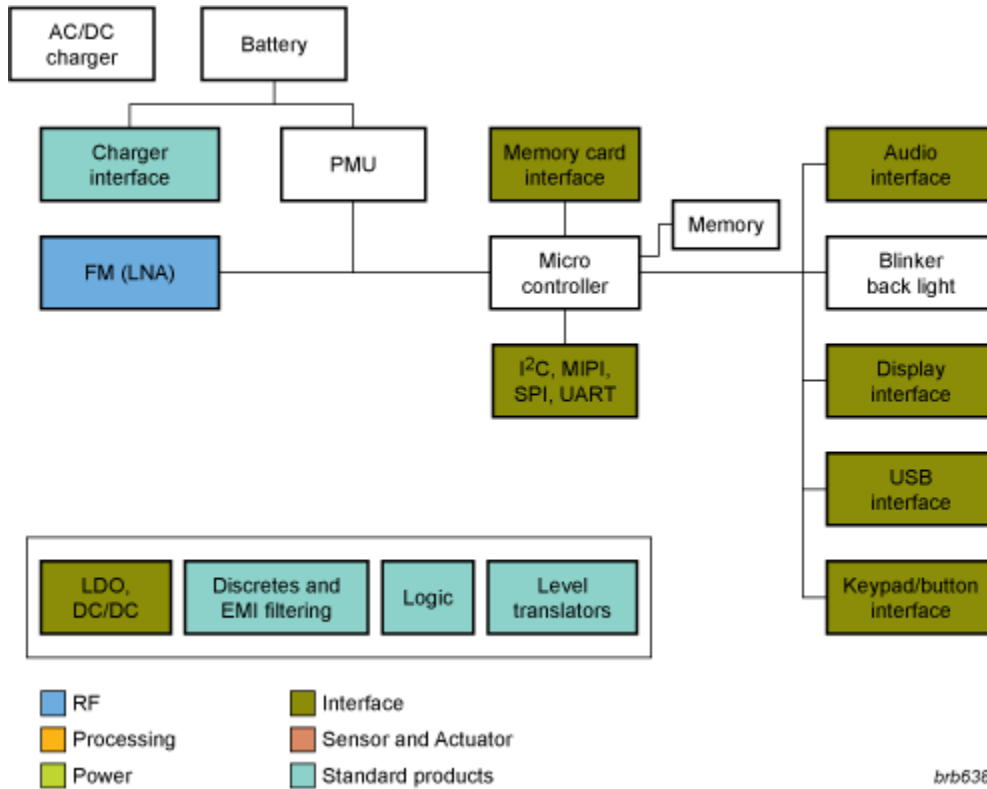


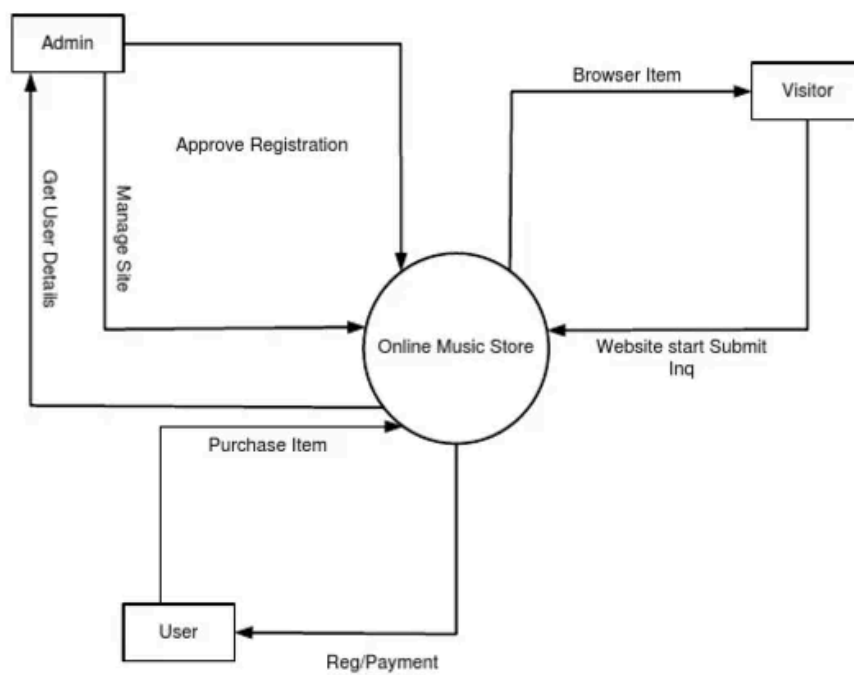
Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

4.2 Interfaces (if applicable)



❖ Data Flow Diagram

Context Level or 0 Level



Performance Test

Performance testing is crucial to demonstrate the real-world feasibility of our music player. Key constraints identified include **memory usage**, **processing speed (MIPS)**, **database performance**, and **user interface responsiveness**.

- **Memory Management:** Optimized by using efficient data structures and proper garbage collection to prevent memory leaks.
- **Processing Speed:** JDBC queries and indexing techniques were used to enhance retrieval speed, ensuring quick access to playlists and songs.
- **Database Performance:** SQL queries were optimized for fast execution, avoiding redundant data fetches to improve efficiency.
- **User Interface Responsiveness:** Designed using Java UI frameworks, ensuring smooth navigation and minimizing lag.

Test Results & Observations

- Playback operations (play, pause, skip) performed without noticeable delay.
- Playlists loaded efficiently with minimal query execution time.
- Database interactions were stable, with no major latency issues.

Future Recommendations

- Implement **multi-threading** for smoother UI performance.
- Use **caching mechanisms** to reduce repetitive database queries.
- Optimize **power consumption** by reducing unnecessary background processing..

4.3 Test Plan/ Test Cases - The testing process ensures the reliability and efficiency of the music player. Key test cases include:

1. **Music Playback** – Verify play, pause, stop, and skip functionalities.
2. **Playlist Management** – Test creation, modification, and deletion of playlists.
3. **Database Operations** – Check song retrieval, playlist storage, and query execution.
4. **UI Responsiveness** – Ensure smooth navigation and user interactions.
5. **Error Handling** – Validate exception handling for invalid inputs and connectivity issues.

4.4 Test Procedure - Execute each feature manually and log the results.

- 4.5 Use sample databases for testing playlist and song retrieval efficiency.
- 4.6 Measure UI response time and detect performance bottlenecks.
- 4.7 Simulate error scenarios (e.g., missing files, database disconnection) to ensure robust handling.

4.8 Performance Outcome

Performance Outcome

- Smooth playback with minimal delays.
- Fast database queries with optimized SQL execution.
- UI remains responsive even with large playlists.
- No critical crashes or unhandled exceptions detected.

My Learnings

Through this project, I gained hands-on experience in Java, JDBC, and database management. I learned how to design and implement a functional music player with playlist management. Debugging errors, optimizing queries, and handling user interactions improved my problem-solving skills. This experience has strengthened my understanding of real-world software development, which will be valuable for future projects and career growth in software engineering.

Future Work Scope

In the future, this project can be enhanced by adding features like online music streaming, AI-based song recommendations, user authentication, and a mobile-friendly UI. Integration with cloud storage for playlist backup and multi-device sync can also be explored. Optimizing performance and implementing advanced audio effects could further improve the user experience.

