

Project Report
Guidelines



MANIPAL UNIVERSITY JAIPUR

University under Section 2(f) of the UCG Act

NAME	Jagdeep Kumar
ROLL NUMBER	2114103301
SEMSETER	6TH
COURSE	BCA
Project Name	WWW.SURBAJAO.COM



Your Ringtone Is Your Signature .

Project Report

Table of Contents:

1. Introduction/Objectives
2. System Analysis
 - Identification of Need
 - Preliminary Investigation
 - Feasibility Study
 - Project Planning
 - Project Scheduling (PERT Chart & Gantt Chart)
3. Software Requirement Specifications (SRS)
 - Software Engineering Paradigm Applied
 - Data Models (DFD, ERD)
 - Modularisation Details
4. System Design
 - Database/Data Security Implementation
 - Creation of User Profiles
 - Managing User Rights
5. Coding
 - SQL Commands
 - Standardization of the Coding
 - Code Efficiency
 - Error Handling
 - Parameters Calling/Passing
 - Validation Checks
6. Testing
 - Testing Techniques and Strategies Used
 - Testing Plan Used
 - Test Reports for Unit Test Cases & System Test Cases
 - Debugging and Code Improvement
7. System Security Measures
 - Database/Data Security Implementation
 - Creation of User Profiles
 - Managing User Rights
8. Cost Estimation and its Model
9. Future Scope
10. Bibliography
11. Appendices (if any)

12. Data Dictionary

❖ **Introduction/Objectives**

In today's world, personalization is key to enhancing user experience across various digital platforms. With the widespread use of smartphones, users seek unique ways to personalize their devices, including setting custom ringtones. However, finding high-quality and diverse ringtones can be challenging, leading to the need for a platform that offers a wide range of options to cater to individual preferences.

The objective of Surbajao is to address this need by providing a comprehensive platform where users can discover, download, and set personalized musical ringtones effortlessly. By leveraging technologies such as PHP, JavaScript, Laravel framework, and

MySQL database, Surbajao aims to offer a user-friendly interface, robust backend functionality, and secure data management.

Key objectives of Surbajao include:

1. Providing a diverse collection of high-quality musical ringtones across various genres and artists.
2. Implementing secure user authentication and profile management features.
3. Facilitating easy searching, previewing, and downloading of ringtones.
4. Ensuring data security and integrity through encryption, hashing, and access control measures.
5. Creating an intuitive and responsive user interface for seamless navigation and interaction.
6. Offering scalability and reliability through cloud hosting solutions.

Surbajao endeavors to become the go-to destination for users seeking personalized musical ringtones, enhancing their smartphone experience and satisfaction.

❖ **System analysis :-** Musical Ringtone Platform

• **Identification of Need:**

- a. **Demand for Musical Ringtones:** There is a growing demand for unique and high-quality musical ringtones among smartphone users who seek to personalize their devices.
- b. **Limited Availability:** Current market offerings lack variety and often fail to meet the diverse musical preferences of users, highlighting the need for a specialized platform.

• **Preliminary Investigation:**

- a. **Market Research:** Conduct surveys, interviews, and market analysis to understand user preferences, existing competitors, and market trends related to musical ringtones.

- b. **Technical Feasibility:** Assess the feasibility of developing a platform that can efficiently host and deliver a wide range of musical ringtones to users across different devices and operating systems.
- c. **Financial Feasibility:** Estimate the initial investment required for development, content acquisition, marketing, and ongoing operational costs, considering potential revenue streams and profitability.

- **Feasibility Study:**

- a. **Technical Feasibility:** Surbajao's technical feasibility is viable given the availability of cloud-based infrastructure and digital audio technologies to host, manage, and deliver musical content to users.
- b. **Economic Feasibility:** The economic feasibility is positive, considering the potential revenue streams from premium downloads, subscriptions, and advertising, which can offset development and operational costs.
- c. **Operational Feasibility:** Surbajao can be operationally feasible with efficient content curation, user-friendly interface design, and scalability to accommodate growing user demand.

- **Project Planning:**

- a. **Scope Definition:** Define the scope of Surbajao, including its features, target audience, content categories, and platform compatibility.
- b. **Resource Identification:** Identify the resources required for development, including human resources (developers, designers), technological resources (servers, software), and financial resources.
- c. **Risk Assessment:** Identify potential risks such as copyright issues, technical challenges, and market competition, and develop strategies to mitigate them.

- **Project Scheduling:**

- a. **PERT Chart:** Develop a Program Evaluation and Review Technique (PERT) chart to outline the sequential tasks involved in Surbajao's development, their dependencies, and the estimated time required for completion.
- b. **Gantt Chart:** Create a Gantt chart to visualize the project timeline, milestones, and resource allocation, facilitating effective project management and tracking of progress.

❖ Software Requirement Specifications (SRS):-

• Software Engineering Paradigm:

For the development of Surbajao, an Agile software development methodology will be applied. Agile allows for iterative and incremental development, enabling flexibility to adapt to changing requirements and feedback from users. This approach ensures rapid delivery of features and maintains a focus on customer satisfaction throughout the development process.

• Data Models:

Entity Relationship Diagram (ERD): An ERD will illustrate the relationships between various entities such as Users, Ringtones, Artists, and Genres. This diagram will help in designing the database structure for Surbajao.

Control Flow Diagram: Control flow diagrams will demonstrate the flow of control within the system, outlining how users navigate through different pages and features of the website.

Sequence Diagrams: Sequence diagrams will depict the interactions between users and the system components, illustrating the sequence of actions performed when downloading and playing a ringtone.

Use-case Diagrams: Use-case diagrams will outline the different user roles and the interactions between users and the system. Use-cases such as "Search for a Ringtone," "Download a Ringtone," and "Play a Ringtone" will be identified.

Activity Diagrams: Activity diagrams will illustrate the workflow of various processes within the system, such as the process of listing ringtones, conducting a search, and downloading a ringtone.

Software Requirements:

Website:

-Landing Page: The landing page will serve as the initial point of entry for users and will provide an overview of Surbajao's features and offerings.

- Ringtones Listing: This page will display a list of available ringtones, categorized by genres or artists, allowing users to browse and explore the selection.

- Search Feature: Users will be able to search for specific ringtones based on criteria such as artist name, genre, or title, enhancing the usability and navigation of the platform.

- Download and Play Ringtone: Users will have the ability to preview and download ringtones directly from the website. Additionally, they can play the selected ringtone before downloading to ensure it meets their preferences.

❖ System Design :-

Modularization Details:

The platform follows the Model-View-Controller (MVC) architecture, which promotes the separation of concerns and modular design. The system is divided into three main modules:

1.Model (M): Handles the business logic and interacts with the database. Includes modules for user management, ringtone management, search functionality, and authentication.

2. View (V): Responsible for presenting the user interface to the users. Includes modules for rendering the landing page, listing ringtones, search results, and download/playback functionalities.

3. Controller (C): Acts as an intermediary between the Model and View, handling user requests and coordinating the flow of data. Includes modules for routing requests, processing user input, and managing session data.

Data Integrity and Constraints:

- **User Data:** Ensures uniqueness of user accounts using unique constraints on email addresses or usernames.
- **Ringtone Data:** Enforces constraints on ringtone metadata such as title, artist, and genre to maintain data integrity.
- **Authentication:** Implements secure authentication mechanisms to prevent unauthorized access and ensure data security.

Database Design Procedural Design/Object Oriented Design:

The database design for Surbajao includes tables for storing user information, ringtone metadata, and session data. It follows normalized database schema to minimize redundancy and maintain data consistency.

- **User Table:** Contains fields such as user ID, username, email, password hash, and account status.
- **Ringtone Table:** Stores information about each ringtone, including ID, title, artist, genre, file path, and download count.
- **Session Table:** Stores session tokens for authenticated users to maintain login state.

Procedural Design/Object Oriented Design:

The procedural design involves breaking down system functionalities into procedures or functions, while Object-Oriented Design (OOD)

focuses on encapsulating data and behavior within objects. Surbajao utilizes a combination of both:

- **Procedural Design:** Handles tasks such as user authentication, data retrieval, and file management using procedural programming techniques.
- **Object-Oriented Design:** Encapsulates entities such as User, Ringtone, and Session as objects with properties and methods, promoting modularity, reusability, and maintainability.

User Interface Design:

The user interface design of Surbajao focuses on simplicity, intuitiveness, and responsiveness. It includes:

- **Landing Page:** Features a visually appealing layout with easy navigation and highlights key features of the platform.
- **Ringtones Listing:** Presents a grid or list view of available ringtones with options for sorting and filtering by genre, artist, or popularity.
- **Search Feature:** Includes a search bar with auto-suggestions and filters to help users quickly find desired ringtones.
- **Download and Playback:** Provides clear call-to-action buttons for downloading and playing ringtones, along with playback controls for previewing before download.

Test Cases:

Unit Test Cases:

1. Test user authentication process.
2. Test ringtone listing functionality.
3. Test search feature with various search queries.
4. Test download functionality for a ringtone.
5. Test playback functionality for a ringtone.

System Test Cases:

1. Test the overall user experience from landing page to downloading a ringtone.
2. Test compatibility across different web browsers and devices.

3. Test system performance under varying user loads.
4. Test error handling and recovery mechanisms.
5. Test security features such as authentication and data encryption.

❖ Coding

1. SQL Commands:

-- Database Creation

```
CREATE DATABASE IF NOT EXISTS Surbajao;
```

-- Use the created database

```
USE Surbajao;
```

-- Users Table

```
CREATE TABLE IF NOT EXISTS Users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  username VARCHAR(50) NOT NULL UNIQUE,  
  email VARCHAR(100) NOT NULL UNIQUE,  
  password VARCHAR(255) NOT NULL,  
  account_status ENUM('Active', 'Inactive') DEFAULT 'Active'  
);
```

-- Ringtones Table

```
CREATE TABLE IF NOT EXISTS Ringtones (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  title VARCHAR(100) NOT NULL,  
  artist VARCHAR(100),  
  genre VARCHAR(50),  
  file_path VARCHAR(255) NOT NULL,  
  download_count INT DEFAULT 0  
);
```

2. Standardization of the Coding:

- Follow Laravel's coding conventions and PSR standards for PHP.

- Use consistent naming conventions for variables, functions, and classes.
- Maintain proper indentation and formatting for better readability.

3. Code Efficiency:

- Utilize Laravel's built-in features like Eloquent ORM for efficient database operations.
- Optimize database queries by eager loading related models and using query builder for complex queries.
- Cache frequently accessed data using Laravel's caching mechanisms.

4. Error Handling:**

- Leverage Laravel's exception handling to gracefully handle errors and exceptions.
- Implement custom error messages and logging for debugging purposes.
- Validate user inputs using Laravel's validation rules to prevent data inconsistencies.

5. Parameters Calling/Passing:

- Use Laravel's route parameters and request object to access and pass parameters securely.
- Sanitize and validate user inputs using Laravel's FormRequest class before processing.

6. Validation Checks:

- Implement validation checks using Laravel's validation rules in controllers or form request classes.
- Validate user authentication and authorization to ensure secure access to resources.
- Apply database constraints and Laravel's validation rules to enforce data integrity.

❖ Testing :

Testing Techniques and Strategies Used:

- **Black-box Testing:** Assessing functionality without knowledge of internal code structures.
- **White-box Testing:** Examining internal code structures to ensure they meet specifications.
- **Functional Testing:** Ensuring each feature meets specified requirements.
- **Regression Testing:** Verifying that recent changes haven't negatively impacted existing functionalities.
- **Usability Testing:** Evaluating user interaction for ease of use.
- **Integration Testing:** Verifying the interaction between integrated components.

- **Performance Testing:** Assessing system performance under various conditions.
- **Security Testing:** Identifying and resolving security vulnerabilities.

Testing Plan Used:

1. Objective: Ensure all functionalities work as intended and meet user requirements.

2. Scope: Cover all features, including user authentication, ringtone listing, search, download, and playback.

3. Resources: Testing team, testing environment, and necessary tools for automation and manual testing.

4. Schedule: Define timelines for each testing phase, including unit testing, system testing, and user acceptance testing.

Test Reports for Unit Test Cases & System Test Cases:

- **Unit Test Report:** Documents test cases and results for individual components.
 - **System Test Report:** Summarizes test cases and results for integrated system functionalities.
- Both reports include details on test scenarios, pass/fail status, identified issues, and resolutions.

Debugging and Code Improvement:

- **Debugging:** Identifying and resolving issues found during testing by analyzing logs and error messages.
 - **Code Improvement:** Enhancing code quality, readability, and efficiency based on feedback from testing.
- Iterative cycles of testing, debugging, and code improvement ensure a stable and reliable Surbajao platform.

❖ System Security Measures:-

• Database/Data Security Implementation:

- Encryption:** Implement encryption techniques such as SSL/TLS to encrypt data transmitted between the server and clients.
- Hashing Passwords:** Store user passwords securely using one-way hashing algorithms like bcrypt to prevent unauthorized access to plaintext passwords.
- Parameterized Queries:** Use parameterized queries or prepared statements to prevent SQL injection attacks and sanitize user inputs.

- d. **Access Control:** Restrict database access to authorized users only and implement role-based access control (RBAC) to manage permissions.

• **Creation of User Profiles:**

- a. **User Authentication:** Implement secure user authentication mechanisms, including email/password authentication, two-factor authentication (2FA), or OAuth for third-party authentication.
- b. **Profile Management:** Allow users to create and manage their profiles with features such as profile picture upload, personal information update, and password reset options.
- c. **Session Management:** Manage user sessions securely using techniques like session tokens, session expiry, and secure session storage to prevent session hijacking.

• **Managing User Rights:**

- a. **Role-Based Access Control (RBAC):** Define roles such as admin, moderator, and regular user, and assign specific permissions to each role to control access to features and data.
- b. **Access Control Lists (ACLs):** Implement ACLs to specify granular access permissions for individual users or user groups based on their roles or attributes.
- c. **Audit Trails:** Maintain audit logs to track user actions and changes to sensitive data, allowing administrators to monitor user activities and detect any unauthorized access or suspicious behavior.

❖ **Cost Estimation and its Model.:-**

Cost estimation for Surbajao involves considering expenses related to development, hosting, maintenance, and marketing. The model used can be a combination of bottom-up estimation, where individual components are estimated and aggregated, and top-down estimation, where overall project costs are estimated based on historical data or expert judgment.

❖ **Future Scope:**

- Integration with cloud storage services for users to upload and use custom ringtones.
- Implementation of social sharing features to allow users to share their favorite ringtones with friends.
- Expansion of platform to support additional file formats and multimedia content.
- Introduction of subscription-based premium features for enhanced user experience.
- Incorporation of machine learning algorithms for personalized recommendations based on user preferences and usage patterns.

❖ **Bibliography:**

- Laracasts. (n.d.). Laravel From Scratch. Retrieved from <https://laracasts.com/series/laravel-8-from-scratch>

- MySQL Documentation. (n.d.). MySQL 8.0 Reference Manual. Retrieved from <https://dev.mysql.com/doc/refman/8.0/en/>
- DigitalOcean Documentation. (n.d.). DigitalOcean Product Documentation. Retrieved from <https://www.digitalocean.com/docs/>

❖ Appendices:

- a. Appendix A: Mockups of User Interface Design
- b. Appendix B: Detailed Project Timeline

❖ Data Dictionary:

1. Users Table:

id: Unique identifier for the user (Primary Key)

username: Username chosen by the user (Unique)

email: Email address of the user (Unique)

password: Hashed password of the user

account_status: Status of the user account (e.g., Active, Inactive)

2. Ringtones Table:

id: Unique identifier for the ringtone (Primary Key)

title: Title of the ringtone

artist: Artist name of the ringtone

genre: Genre of the ringtone

file_path: File path to the ringtone file

download_count: Number of downloads for the ringtone

This data dictionary provides a clear definition of the fields present in the Users and Ringtones tables, facilitating understanding and management of the database schema.