ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 4 HND Diploma in Computing**		
Unit number and title	Unit 7: Software Development Life Cycle		
Submission date	11/26/2023	**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Lai Van Cuong	**Student ID**	BH01311
Class	IT0604	**Assessor name**	Dinh Van Dong

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

		Student's signature	Cuong

Grading grid

P5	P6	P7	M3	M4	M5	M6	D3	D4

❐ Summative Feedback:

❐ Resubmission Feedback:

Grade:

Assessor Signature:

Date:

Internal Verifier's Comments:

Signature & Date:

Table of content

Table of Contents

Table of figures

Introduction

Tune Source is a company headquartered in southern California. Tune Source is the brainchild of three entrepreneurs with ties to the music industry: John Margolis, Megan Taylor, and Phil Cooper. Originally, John and Phil partnered to open a number of brick-and-mortar stores in southern California specializing in hard-to-find and classic jazz, rock, country, and folk recordings. Megan soon was invited to join the partnership because of her contacts and knowledge of classical music. Tune Source quickly became known as the place to go to find rare audio recordings. Annual sales last year were $40 million with annual growth at about 3%–5% per year. Tune Source currently has a website that enables customers to search for and purchase CDs. This site was initially developed by an Internet consulting firm and is hosted by a prominent local Internet Service Provider (ISP) in Los Angeles. The IT department at Tune Source has become experienced with Internet technology as it has worked with the ISP to maintain the site.

Project Sponsor: Carly Edwards, Assistant Vice President, Marketing.

Business Need: This project has been initiated to increase sales by creating the capability of selling digital music downloads to customers through kiosks in our stores, and over the Internet using our website.

Business Requirements: Using the Web or in-store kiosks, customers will be able to search for and purchase digital music downloads. The specific functionality that the system should have includes the following:

- Search for music in our digital music archive.
- Listen to music samples.
- Purchase individual downloads at a fixed fee per download.
- Establish a customer subscription account permitting unlimited downloads for a monthly fee.
- Purchase music download gift cards.

Business Value: We expect that Tune Source will increase sales by enabling existing customers to purchase specific digital music tracks and by reaching new customers who are interested in our unique archive of rare and hard-to-find music. We expect to gain a new revenue stream from customer subscriptions to our download services. We expect some increase in cross-selling, as customers who have downloaded a track or two of a CD decide to purchase the entire CD in a store or through our website. We also expect a new revenue stream from the sale of music download gift cards.

Conservative estimates of tangible value to the company include the following:

- $757,500 in sales from individual music downloads.
- $950,000 in sales from customer subscriptions.
- $205,000 in additional in-store or website CD sales.
- $153,000 in sales from music download gift cards.

Special Issues or Constraints:

- The marketing department views this as a strategic system. The ability to offer digital music downloads is critical in order to remain competitive in our market niche. Our music archive of rare and hard-to-find music is an asset that is currently underutilized.
- Many of our current loyal customers have been requesting this capability, and we need to provide this service or face the loss of these customers' business.'
- Because customers have a number of music download options available to them elsewhere, we need to bring this system to the market as soon as possible.

Content

P5 Undertake a software investigation to meet a business need.

Identify the stakeholders, their roles, and interests:

Founders (John Margolis, Phil Cooper, and Megan Taylor):

- **Roles:** Founders deeply invested in Tune Source.
- **Interests:** Company profitability, growth, and maintaining a reputation for rare and hard-to-find music.

Carly Edwards (Project Sponsor - Assistant Vice President, Marketing):

- **Role:** Oversees the project, ensures alignment with marketing strategies.
- **Interest:** Successful implementation of digital music downloads to boost sales and customer engagement.

IT Department:

- **Role:** Responsible for technical implementation, development, and maintenance of the website and kiosks.
- **Interest:** Delivering a technically sound solution meeting business requirement.

Customers:

- **Role:** Customer buying products at Tune Source, the hard-to-find music.
- **Interest:** Existing customers desire user-friendly access to purchase digital music. New customers attracted by the unique archive and subscription offerings.

Marketing Department:

- **Role:** Views the project strategically, aiming to leverage the unique archive to gain a competitive edge.
- **Interest:** Views the project strategically, aiming to leverage the unique archive for a competitive edge. Interested in using digital downloads as a marketing tool.

Functional Requirements (FRs) and Non-Functional Requirements (NFRs) of Tune Source Project:

What are functional requirements?

Functional requirements (FRs) define the specific functionalities, features, and capabilities that a system, software application, or product must provide to its users. These requirements outline what the system is expected to do in terms of user interactions, data manipulation, and overall behavior. Functional requirements are often documented during the early stages of the software development life cycle and serve as a foundation for design, implementation, and testing.

User Interactions:

- Describing how users will interact with the system.
- Specifying the features and actions available to users.

System Behavior:

- Defining the expected behavior of the system in response to user inputs.
- Outlining the processes and tasks the system must perform.

Data Handling:

- Detailing how data is input, processed, stored, and output by the system.
- Identifying data validation and manipulation requirements.

External Interfaces:

- Describing interactions with external systems, devices, or software components.
- Outlining communication protocols and data exchange formats.

Performance Requirements:

- Specifying the system's performance criteria, such as response times and throughput.
- Defining scalability requirements to accommodate potential growth.

Constraints:

- Identifying any limitations or constraints on the system, such as platform compatibility or regulatory compliance.

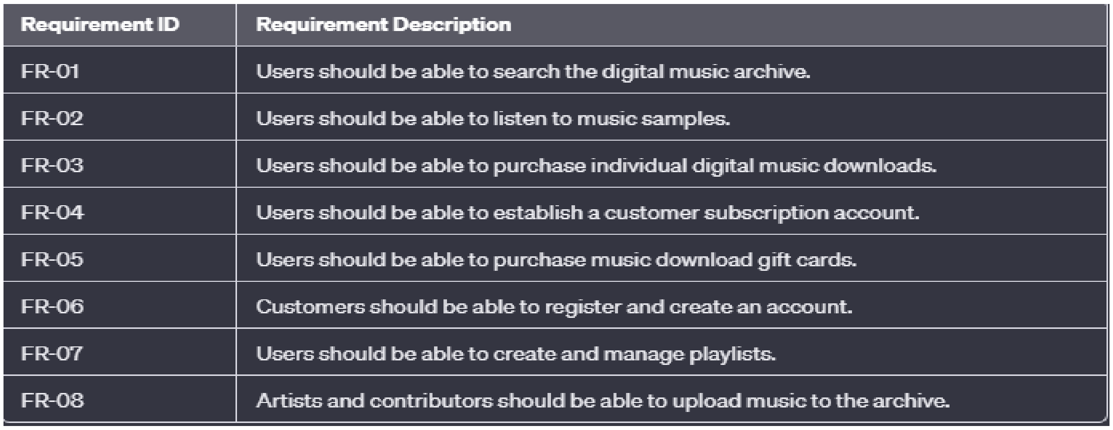
Use Cases and Scenarios:

- Describing specific use cases, user scenarios, and workflows supported by the system.
- Outlining the expected outcomes of different user interactions.

Documentation:

- Defining the documentation requirements for the system, including user manuals, help guides, and technical documentation.

Functional Requirements:



Requirement ID	Requirement Description
FR-01	Users should be able to search the digital music archive.
FR-02	Users should be able to listen to music samples.
FR-03	Users should be able to purchase individual digital music downloads.
FR-04	Users should be able to establish a customer subscription account.
FR-05	Users should be able to purchase music download gift cards.
FR-06	Customers should be able to register and create an account.
FR-07	Users should be able to create and manage playlists.
FR-08	Artists and contributors should be able to upload music to the archive.

Figure 1: Functional Requirement

- **Search for Music:** Users should be able to search the digital music archive effectively.
- **Listen to Music Samples:** Provide the functionality for users to preview music before purchasing.
- **Purchase Individual Downloads:** Enable customers to buy specific digital music tracks at a fixed fee per download.
- **Customer Subscription Account:** Allow customers to subscribe, offering unlimited downloads for a monthly fee.
- **Purchase Gift Cards:** Implement the ability to buy gift cards for digital music downloads.

What are non-functional requirements?

Non-functional requirements (NFRs) specify the criteria that do not directly relate to the specific behaviors or features of a system but are essential for evaluating its overall performance, usability, security, and other qualities. Unlike functional requirements that define what a system should do, non-functional requirements define how well the system should perform those functions. These requirements are often categorized based on various attributes that contribute to the overall quality and effectiveness of the system.

Performance:

Figure 2: Performance

Examples:

- "The system response time for search queries should be within 2 seconds."
- “The platform should support a minimum of 10,000 concurrent users.”

Attributes:

- Response time, throughput, scalability.

Security:

Figure 3: Security

Examples:

- "User data and transactions must be encrypted using industry-standard protocols."
- “Access to certain functionalities should be restricted based on user roles.”

Attributes:

- Data encryption, access control, authentication.

Reliability:

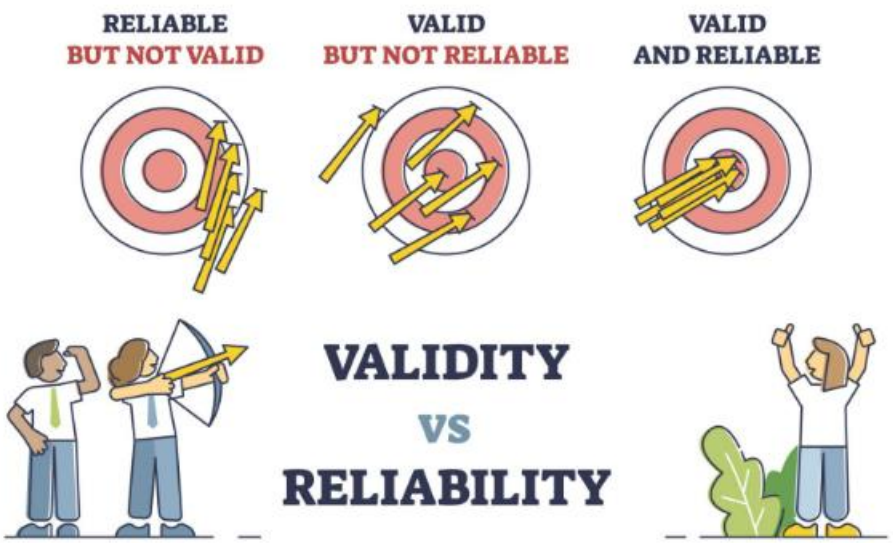


Figure 4: Reliability

Examples:

- "The system should have a 99.9% uptime."
- “Regular backups of the music archive and user data should be performed.”

Attributes:

- Availability, fault tolerance, backup procedures.

Usability:

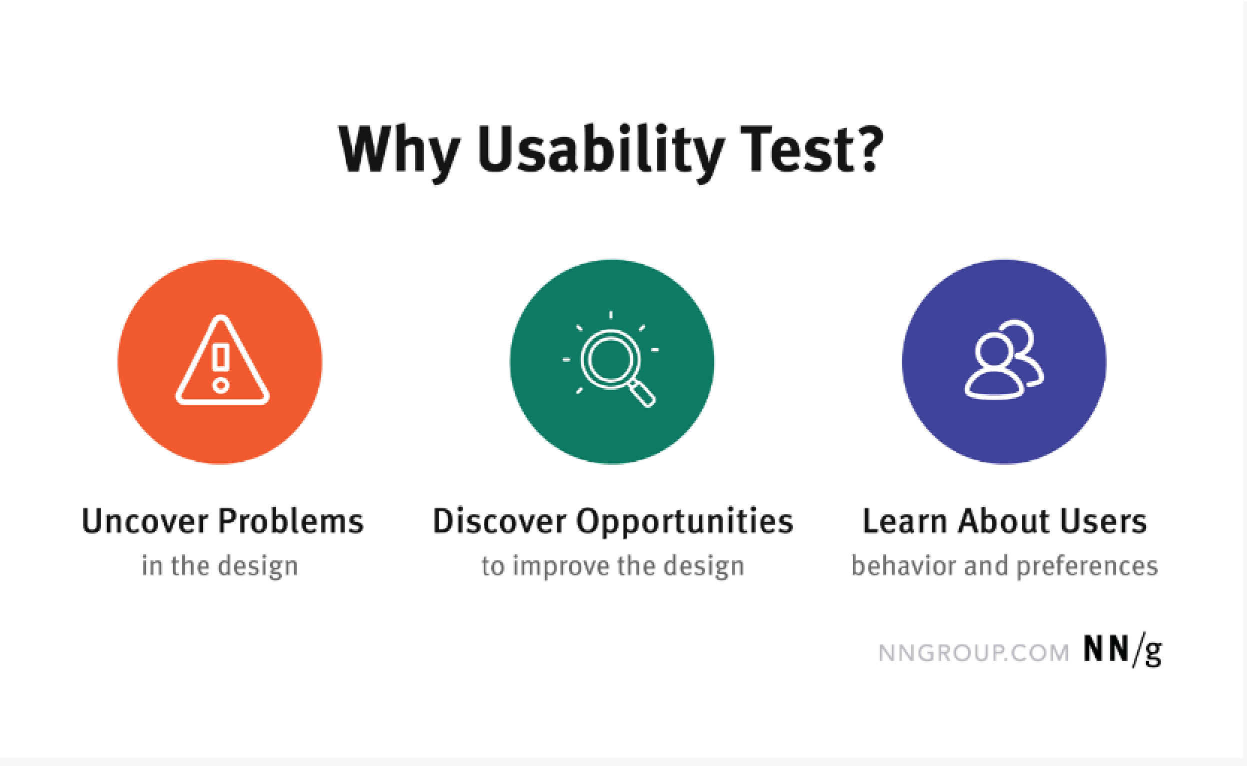


Figure 5: Usability

Examples:

- "The user interface should be intuitive and easy to navigate."
- “The platform should adhere to accessibility standards (WCAG 2.0).”

Attributes:

- User interface design, accessibility, user training.

Compatibility:

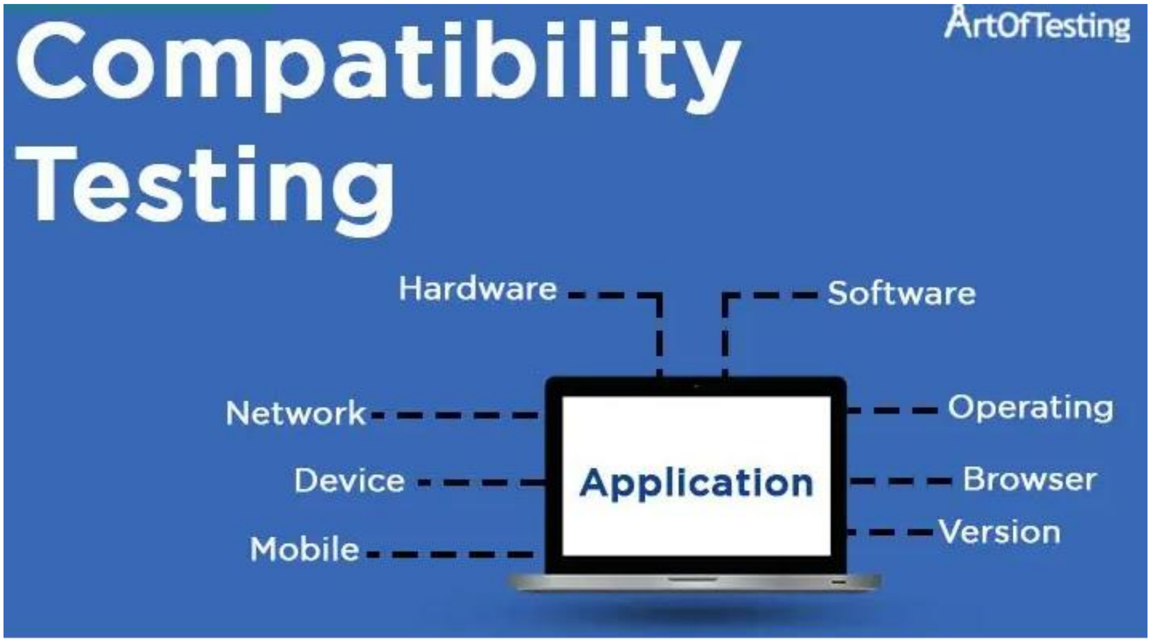


Figure 6: Compatibility

Examples:

- "The platform should be compatible with major browsers (Chrome, Firefox, Safari)."
- “The system should support both desktop and mobile devices.”

Attributes:

- Cross-browser compatibility, cross-device compatibility.

Scalability:

Figure 7: Scalability

Examples:

- "The system should scale horizontally to accommodate increased user loads."
- “Database storage should be scalable to handle a growing music archive.”

Attributes:

- Horizontal scalability, vertical scalability.

Maintainability:

Figure 8: Maintainability

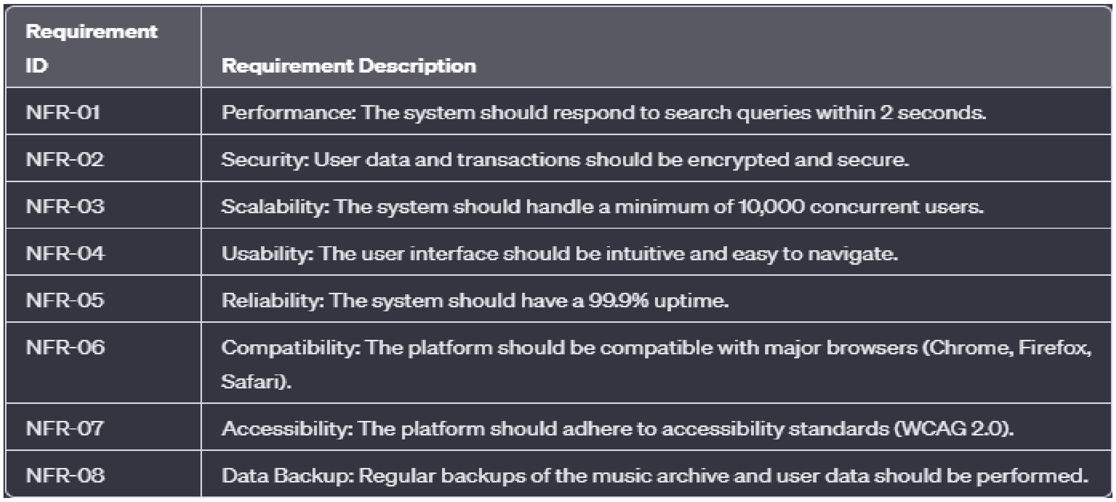
Examples:

- "Code should be well-documented to facilitate future updates and maintenance."
- “System updates should be deployable without disrupting user access.”

Attributes:

- Code readability, documentation, ease of updates.

Non-Functional Requirements:



Requirement ID	Requirement Description
NFR-01	Performance: The system should respond to search queries within 2 seconds.
NFR-02	Security: User data and transactions should be encrypted and secure.
NFR-03	Scalability: The system should handle a minimum of 10,000 concurrent users.
NFR-04	Usability: The user interface should be intuitive and easy to navigate.
NFR-05	Reliability: The system should have a 99.9% uptime.
NFR-06	Compatibility: The platform should be compatible with major browsers (Chrome, Firefox, Safari).
NFR-07	Accessibility: The platform should adhere to accessibility standards (WCAG 2.0).
NFR-08	Data Backup: Regular backups of the music archive and user data should be performed.

Figure 9: Non-functional requirements

- **Performance:** Ensure the system is responsive, handles concurrent users, and delivers music promptly.
- **Security:** Implement robust security measures to protect customer data and prevent unauthorized access.
- **Scalability:** Design the system to handle an increasing number of users and a growing music archive.
- **Usability:** Ensure a user-friendly interface for both the website and in-store kiosks.

Relationships between FRs and NFRs:

The performance requirement is crucial for the seamless functionality of searching, previewing, and purchasing music. Security is fundamental in protecting customer transactions and data during purchases and subscription sign-ups. Scalability is linked to the system's ability to handle an expanding library and increasing user base. Usability directly impacts the success of all functional requirements, ensuring that customers can easily navigate and utilize the digital music platform.

The technique(s) use to obtain these requirements at Tune-source.

The methods used to obtain requirements at Tune Source can vary depending on the specific needs and characteristics of the project. Business Process Analysis (BPA), Business Process Improvement (BPI), and Business Process Reengineering (BPR) are methodologies that can be employed individually or in combination to gather, analyze, and improve requirements.

1. Business Process Analysis (BPA):

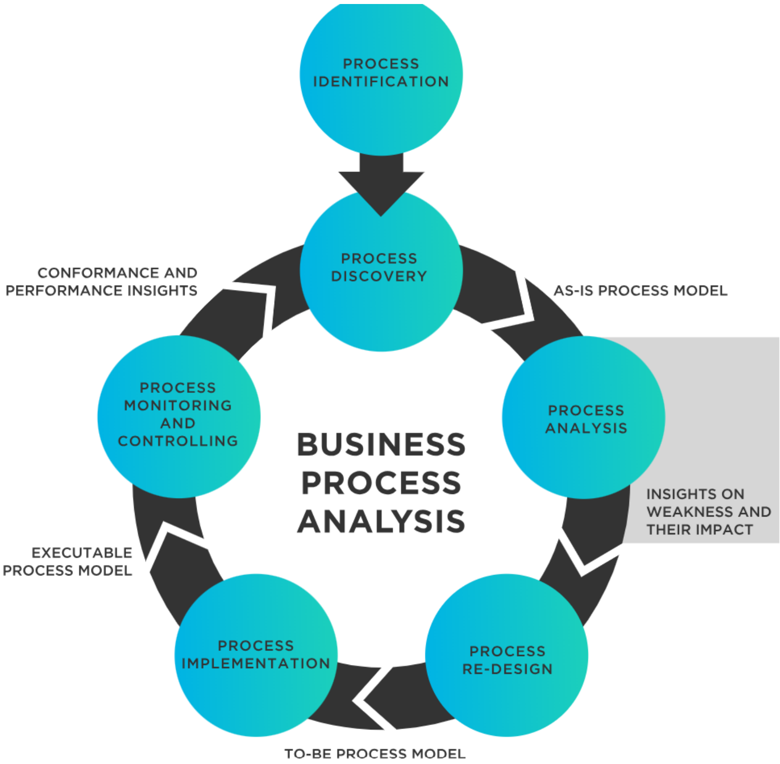


Figure 10: Business Process Analysis

Definition: Business Process Analysis (BPA) is a systematic approach that involves a comprehensive examination of current business processes to gain a deep understanding of their intricacies and to identify potential areas for enhancement.

Application at Tune Source: BPA can be instrumental in assessing Tune Source's current business processes, especially those related to CD sales, customer interactions, and inventory management. Through detailed process mapping and analysis, Tune Source can pinpoint inefficiencies or bottlenecks in its current operations. Specifically, in the context of introducing digital music downloads, BPA can shed light on areas where the integration of this new feature can streamline processes and enhance overall efficiency.

2. Business Process Improvement (BPI):

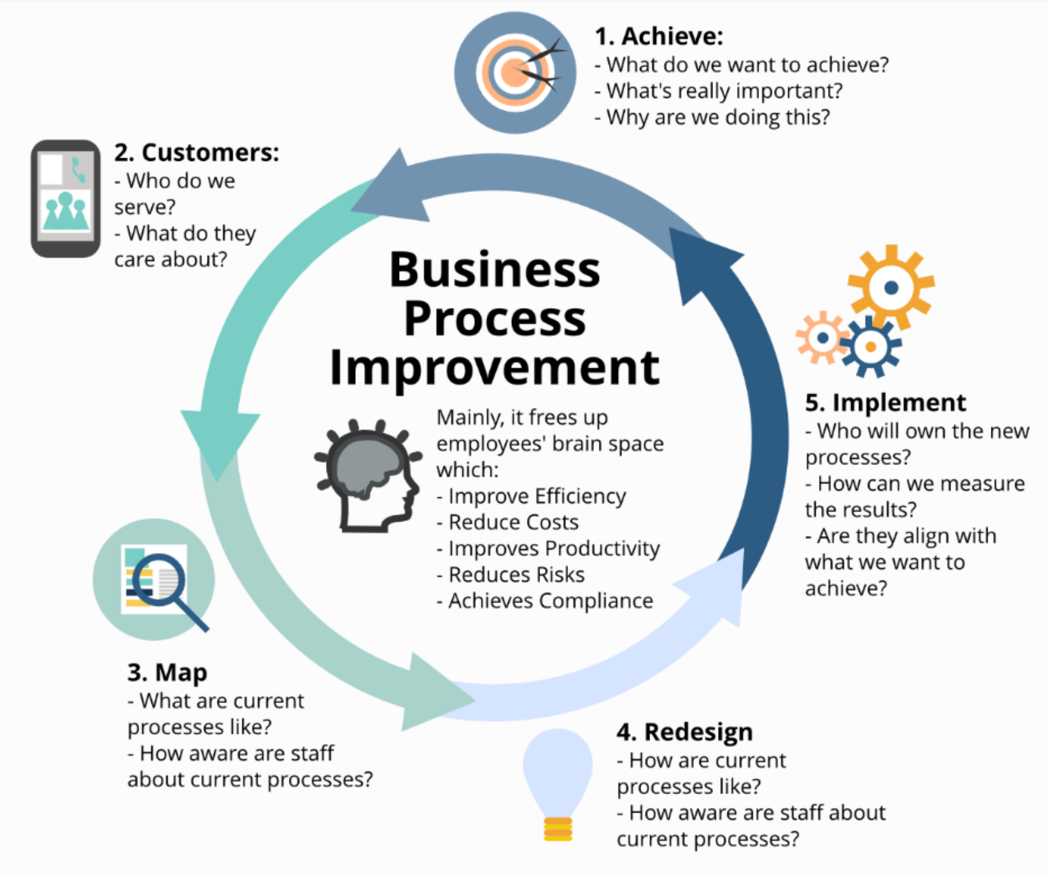

1. Achieve:
- What do we want to achieve?
- What's really important?
- Why are we doing this?
2. Customers:
- Who do we serve?
- What do they care about?
Business Process Improvement
Mainly, it frees up employees' brain space which:
- Improve Efficiency
- Reduce Costs
- Improves Productivity
- Reduces Risks
- Achieves Compliance
5. Implement
- Who will own the new processes?
- How can we measure the results?
- Are they align with what we want to achieve?
3. Map
- What are current processes like?
- How aware are staff about current processes?
4. Redesign
- How are current processes like?
- How aware are staff about current processes?

Figure 11: business Process Improvement

Definition: Business Process Improvement (BPI) focuses on making incremental, gradual changes to existing processes with the aim of enhancing overall efficiency, reducing costs, and improving the quality of outcomes.

Application at Tune Source: BPI is applicable to Tune Source's current CD sales process. By analyzing each step in the sales cycle, BPI can identify opportunities for streamlining order processing, improving inventory management, and enhancing customer interactions. Moreover, within the framework of introducing digital music downloads, BPI can pinpoint specific areas where the integration of this new functionality can be seamlessly woven into existing processes, ensuring a smooth and positive customer experience.

3. Business Process Reengineering (BPR):

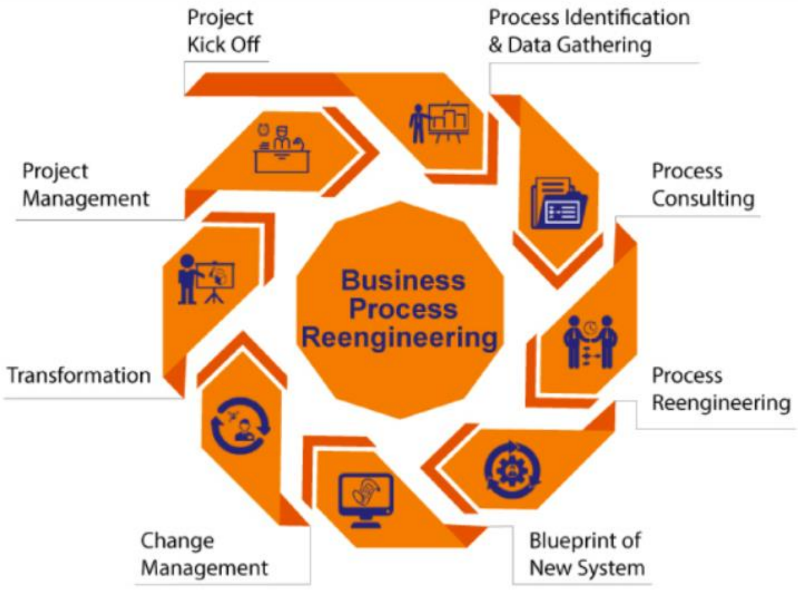


Figure 12:Business Process Reengineering

Definition: Business Process Reengineering (BPR) involves a radical redesign of business processes, often leveraging new technologies, to achieve significant and transformative improvements in performance and efficiency.

Application at Tune Source: BPR could be considered if Tune Source envisions a complete transformation of its business model, with a heightened focus on digital music downloads. This might involve reimagining the entire customer journey, from how Tune Source interacts with its customers to how it manages its extensive music archive. Through BPR, Tune Source can explore innovative ways to deliver digital content, harnessing new technologies and workflows to bring about a revolutionary change in its operational dynamics.

4. Joint Application Development (JAD):

Figure 13: Joint Application Development

The project requires active collaboration among stakeholders for a shared understanding.

Demonstration: Conduct JAD sessions involving key stakeholders—founders, Carly Edwards, and the IT department. The collaborative nature of JAD sessions fosters open communication, allowing stakeholders to collectively define project objectives, functionalities, and potential challenges. The focus will be on eliciting high-level requirements, ensuring alignment with business goals. Additionally, JAD will facilitate the identification of dependencies and trade-offs among different requirements.

Advantages:

- **Collaborative Decision-Making:** Involves key stakeholders in decision-making, fostering collaboration and ensuring diverse perspectives are considered.
- **Efficiency:** Accelerates the requirement gathering process by bringing stakeholders together for focused sessions.

Disadvantages:

- **Resource Intensive:** Requires significant time and commitment from stakeholders, which may be challenging to schedule.
- **Potential for Dominance:** Certain stakeholders may dominate discussions, leading to a skewed representation of requirements.

5. Interviews:

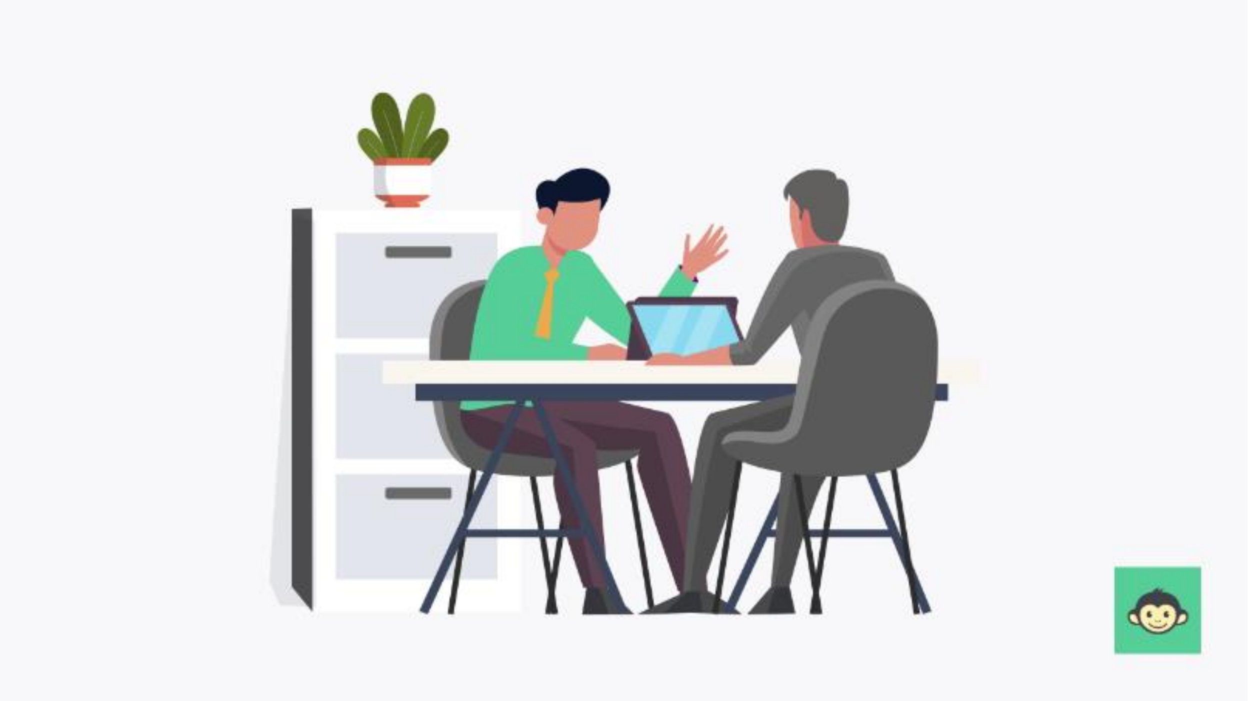


Figure 14: Interviews

Certain requirements may be best captured through one-on-one discussions, especially when understanding individual stakeholder perspectives.

Demonstration: Conduct structured interviews with Carly Edwards, founders, and key IT personnel. These interviews will delve into specific details related to marketing strategies, expectations for digital music downloads, technical considerations, and potential challenges. Open-ended questions will be used to allow stakeholders to express their individual insights, ensuring that unique concerns and aspirations are captured.

Advantages:

- **In-Depth Understanding:** Provides a deep understanding of individual stakeholder perspectives, uncovering nuanced requirements.
- **Flexibility:** Allows for follow-up questions and clarification, adapting to the unique insights of each interviewee.

Disadvantages:

- **Subject to Bias:** Responses may be influenced by personal biases or limited to the interviewee's specific role.
- **Time-Consuming:** Individual interviews can be time-consuming, especially when multiple stakeholders are involved.

6. Observation:

Figure 15: observation

Understanding user behavior and preferences is crucial for designing a user-friendly digital platform.

Demonstration: Observe customers in both brick-and-mortar stores and the existing website. Focus on how customers interact with the current offerings, their preferences in searching for music, and any challenges faced. This observational data will inform the design of the digital platform, ensuring that it aligns with customer expectations and preferences. For instance, observing the in-store experience will reveal insights into how customers navigate physical music collections, guiding the design of the digital kiosk interface.

Advantages:

- **Real-World Insights:** Offers direct insights into user behavior, preferences, and pain points in real-world scenarios.
- **Unbiased Data:** Observations provide unbiased data, revealing actual user interactions without relying on self-reported information.

Disadvantages:

- **Limited Context:** Observations may lack context, requiring additional methods to understand the reasons behind certain behaviors.
- **Observer Effect:** The presence of an observer may influence user behavior, impacting the authenticity of the data.

7. Prototyping:

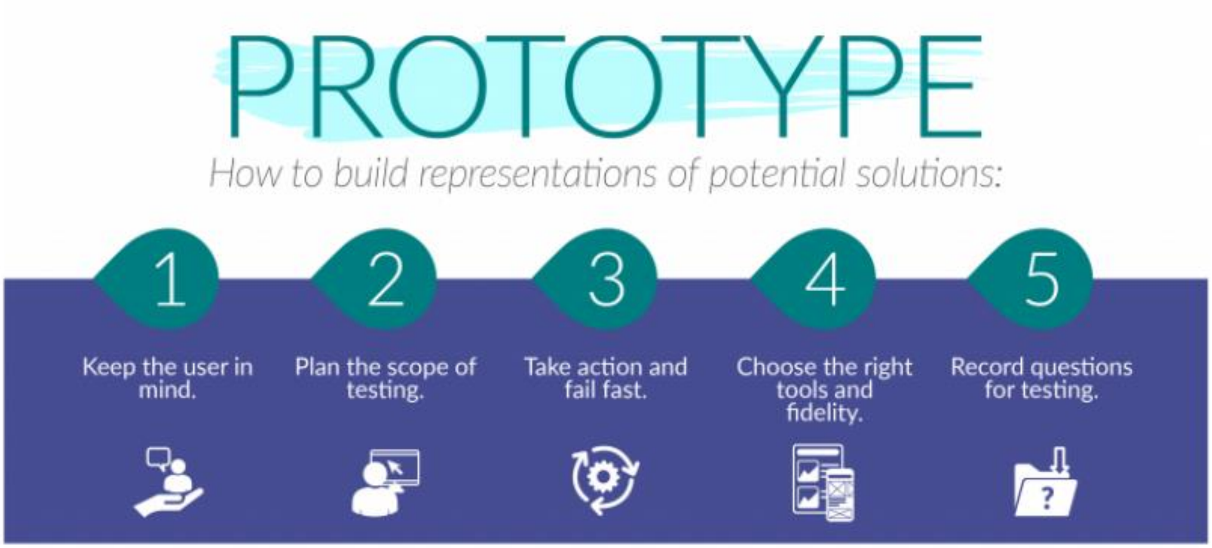

PROTOTYPE
How to build representations of potential solutions:
1
Keep the user in mind.
2
Plan the scope of testing.
3
Take action and fail fast.
4
Choose the right tools and fidelity.
5
Record questions for testing.

Figure 16: prototyping

Iterative feedback is necessary for refining and validating requirements.

Demonstration: Develop prototypes based on the gathered requirements and present them to stakeholders for feedback. This iterative process allows stakeholders to visualize the proposed digital music platform, providing valuable insights into usability, functionality, and design preferences. The prototype serves as a tangible representation, making it easier for stakeholders to provide constructive feedback and refine their requirements based on the evolving design.

Advantages:

- **Visual Representation:** Allows stakeholders to visualize proposed solutions, aiding in clearer communication.
- **Iterative Refinement:** Facilitates an iterative process of refinement based on continuous feedback, reducing the risk of misunderstandings.

Disadvantages:

- **Resource Intensive:** Developing prototypes can be time and resource-intensive.
- **May Not Capture All Requirements:** Visual representations may not capture all intricacies and details of requirements.

8. Surveys:

Figure 17: surveys

Gathering input from a wider audience is essential for understanding diverse customer preferences.

Demonstration: Distribute surveys to existing and potential customers to gather quantitative data on their preferences for digital music downloads. The surveys will include questions about the desired features, pricing structures, and overall satisfaction with the current offerings. The quantitative data obtained will complement the qualitative insights from interviews and observations, providing a more comprehensive understanding of customer needs.

Advantages:

- **Quantifiable Data:** Surveys provide quantifiable data, allowing for statistical analysis of trends and preferences.
- **Scalability:** Can be distributed to a large audience, providing insights from a diverse range of respondents.

Disadvantages:

- **Limited Depth:** Surveys may not capture the depth of qualitative insights obtained through interviews or observations.
- **Low Response Rates:** Response rates can be low, impacting the representativeness of the data.

9. Conclusion:

Given the unique characteristics of Tune Source's project, a combination of methods is recommended. Joint Application Development (JAD) can kickstart the process by involving key stakeholders in collaborative discussions, ensuring a shared vision. Interviews should be used to delve deeper into individual perspectives, especially with Carly Edwards and the founders. Observation is essential to understand user behavior in real-world scenarios, providing insights for designing an intuitive platform. Prototyping allows for iterative refinement based on continuous feedback, ensuring the evolving design aligns with stakeholder expectations. Lastly, Surveys can be distributed to a wider audience for quantifiable data on customer preferences.

This multifaceted approach ensures that Tune Source captures the depth and breadth of requirements, combining the strengths of each method to create a well-rounded understanding. Regular feedback loops, especially through prototyping, will help validate assumptions and refine requirements iteratively, leading to a more successful implementation of the digital music download project.

P6 Use appropriate software analysis tools/techniques to carry out a software investigation and create supporting documentation.

What is Use case Modeling?

Use Case Modeling is a technique in software engineering that involves creating a visual representation of how a system will interact with external entities, known as actors, to accomplish specific goals. It helps in understanding the functional requirements of a system by illustrating various scenarios and interactions between users and the system.

Actors:

- **Customer:** Represents individuals who use the Tune Source platform to search for, listen to, and purchase digital music downloads.
- **Artist/Contributor:** Represents individuals who contribute to the music archive by uploading their music to Tune Source.
- **Administrator:** Represents individuals responsible for managing the overall system, including user accounts, content moderation, and system configuration.

Use Case Diagram for the whole system.

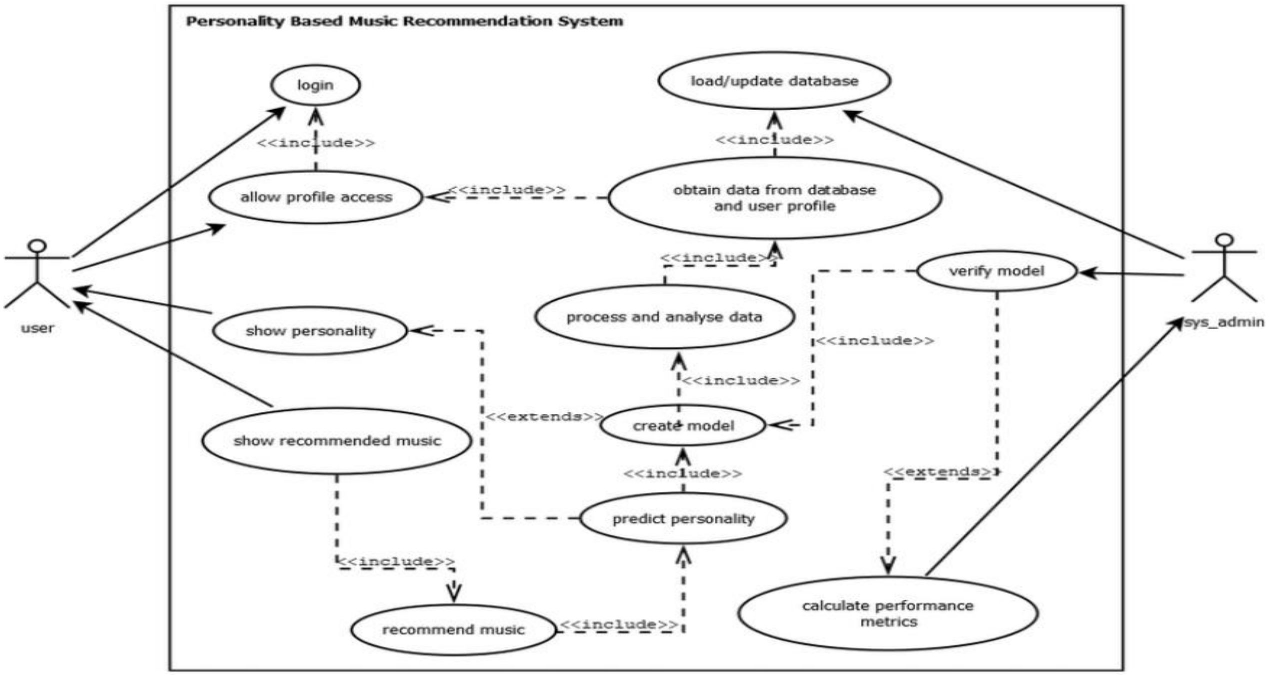


Figure 18: Use Case Diagram

A Use Case Diagram provides a high-level view of the system's functionalities from an external perspective. For Tune Source's digital music download system, key actors include Customers, Admin, and the System itself. Use cases represent actions or services the system provides.

Actors:

- **Customer:** Represents individuals interacting with the system.
- **Admin:** Represents administrative users responsible for system management

Use Cases:

- **Search for Music:** Allows customers to search the digital music archive.
- **Listen to Music Samples:** Permits customers to preview music before purchasing.
- **Purchase Individual Downloads:** Enables customers to buy specific digital music tracks.
- **Customer Subscription Account:** Allows customers to subscribe for unlimited downloads.
- **Purchase Gift Cards:** Facilitates the purchase of gift cards for digital music downloads.
- **Admin Management:** Covers functionalities for system administration and content management.

Use Case specification for 2 Use cases.

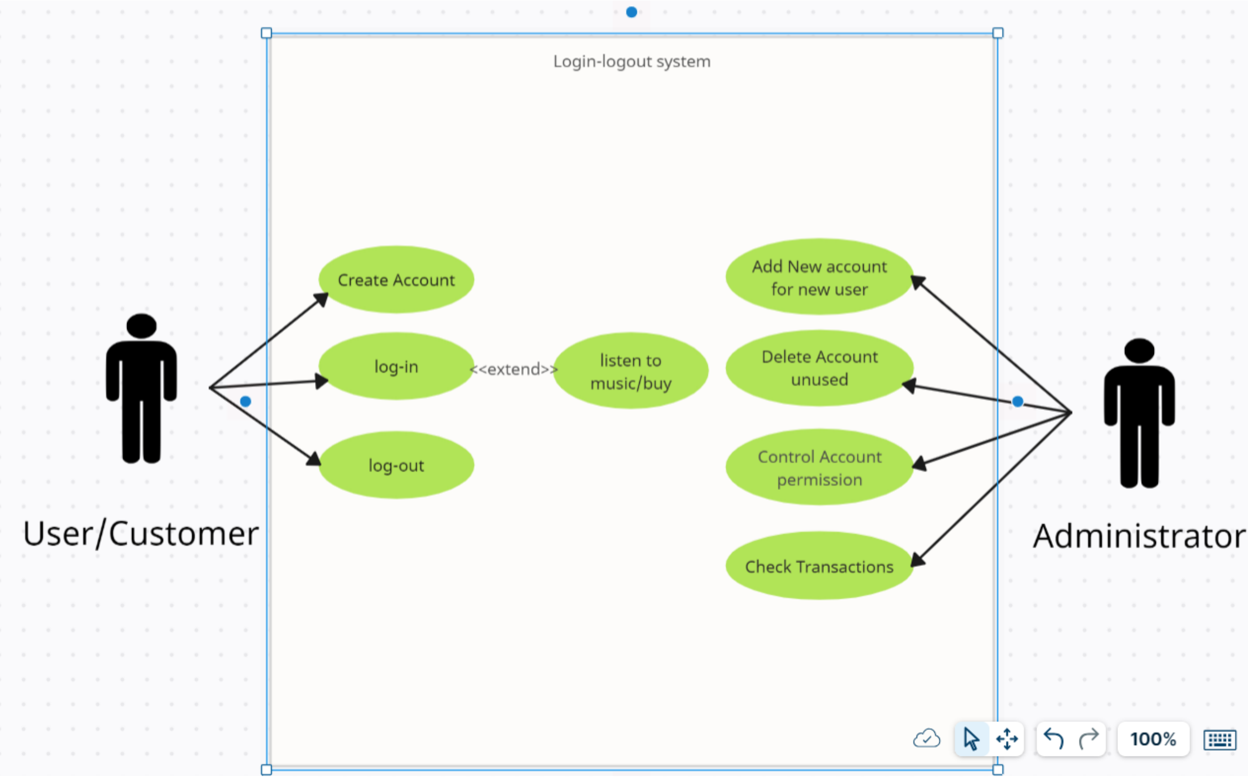


Figure 19: Log-in/ Log-out case specification

Use Case: Log-In

- **Actor:** Customer
- **Description:** This use case enables customers to log into the digital music system.
- **Preconditions:** Customer is not currently logged into the system.

Main Flow:

Customer accesses the login page.

Customer enters their username and password.

System verifies the credentials.

Upon successful verification, the system grants access.

Postconditions: Customer is logged into the system, gaining access to personalized features.

Context Diagram for the whole system.

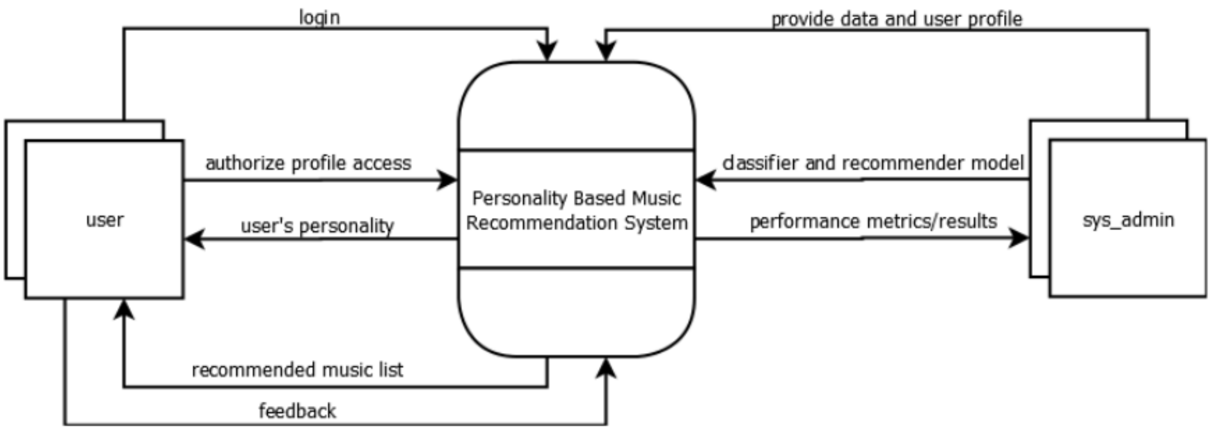

login
provide data and user profile
user
authorize profile access
user's personality
Personality Based Music Recommendation System
classifier and recommender model
performance metrics/results
sys_admin
recommended music list
feedback

Figure 20: Context Diagram

A Context Diagram provides a high-level view of the system and its external entities, illustrating the boundaries of the system.

External Entities:

- **Customers:** Represented as the primary external users interacting with the system.
- **Admin:** Represents administrative users managing the system.

Processes:

- **Digital Music System:** Represents the core system facilitating digital music downloads.

Data Flows:

- **Customer Search Queries:** Flow from Customers to the Digital Music System.
- **Purchase Information:** Flows between Customers and the Digital Music System.

Data Flow Diagram – Level 0 for the whole system.

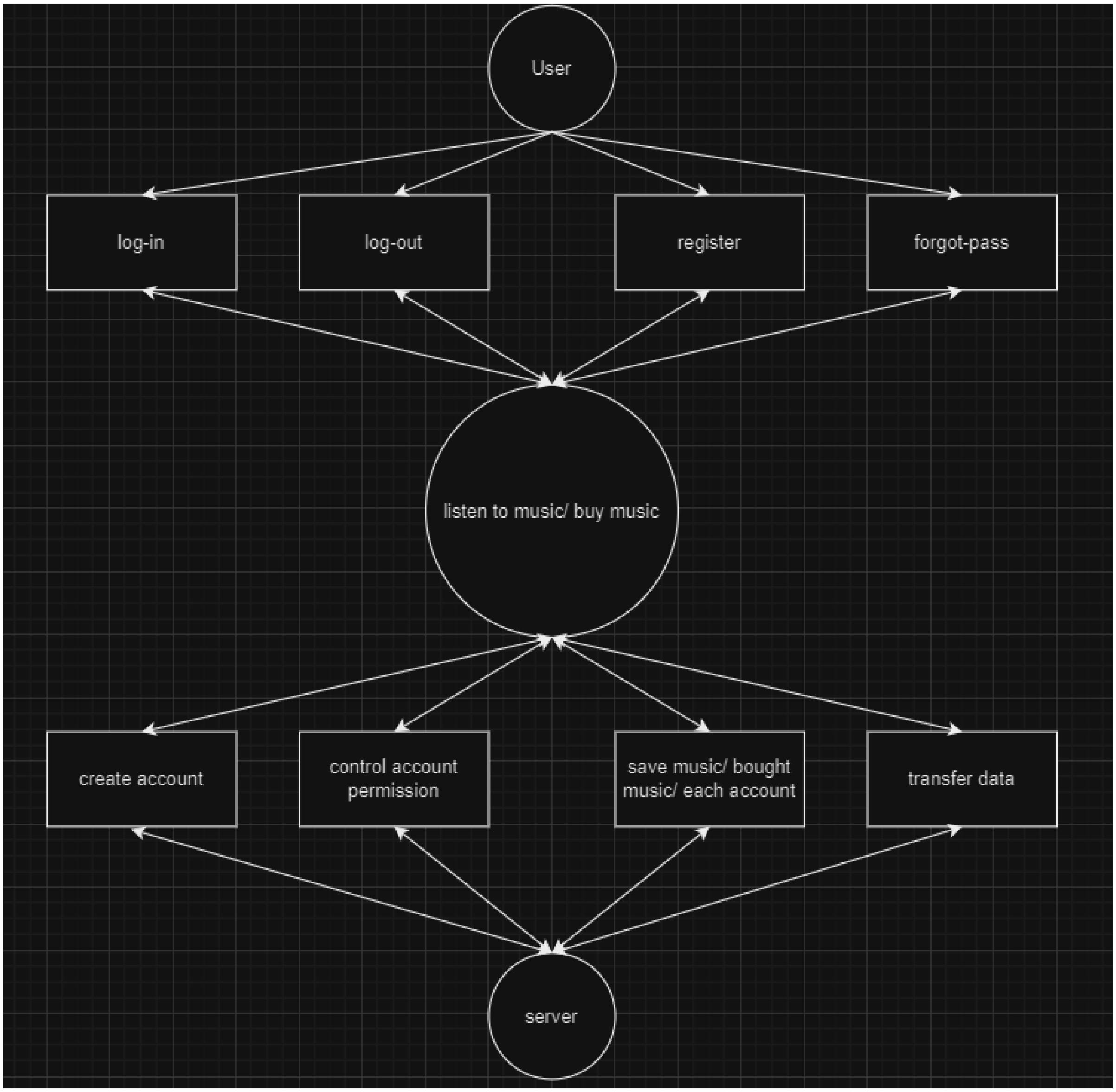

User
log-in
log-out
register
forgot-pass
listen to music/ buy music
create account
control account permission
save music/ bought music/ each account
transfer data
server

Figure 21: Dataflow Diagram

A Level 0 DFD provides an overview of the system's major processes and their interactions.

Processes:

- **Customer Interaction:** Represents all customer-related activities, including search, purchase, and subscription.
- **Admin Management:** Covers tasks related to system administration and content management.

Data Flows:

- **Customer Search Queries:** Flow into the Customer Interaction process.
- **Purchase Information:** Flows into the Customer Interaction process.

Data Stores:

- **Digital Music Archive:** Stores information about available music tracks.
- **Customer Accounts:** Contains customer information, purchase history, and subscription details.

External Entities:

- **Customers:** Represented as the primary external users interacting with the system.
- **Admin:** Represents administrative users managing the system.

ERD for the whole system.

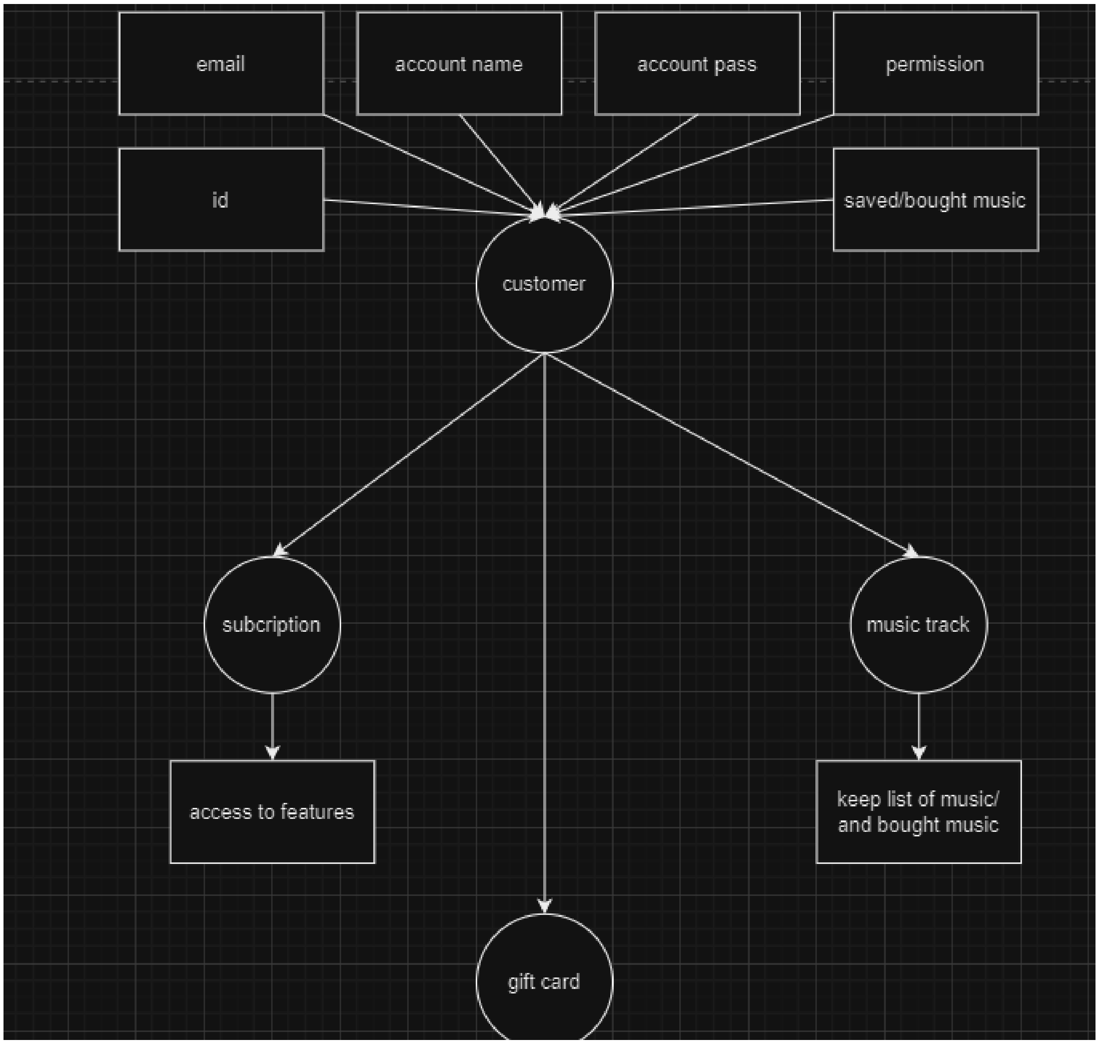


Figure 22: ERD for entities

An ERD illustrates the relationships between entities in a system.

Entities:

- **Customer:** Represents individuals interacting with the system.
- **Music Track:** Represents individual digital music tracks available for purchase.
- **Subscription:** Represents customer subscription details.
- **Gift Card:** Represents digital music download gift cards.

Relationships:

- **Customer searches Music Track:** Indicates the association between customers and the music tracks they search for.
- **Customer purchases Music Track:** Depicts the relationship between customers and the music tracks they buy.
- **Customer subscribes to Subscription:** Represents the connection between customers and subscription services.
- **Customer buys Gift Card:** Illustrates the relationship between customers and purchased gift cards.

Conclusion:

These modeling techniques provide a comprehensive view of Tune Source's digital music download system. The Use Case Diagram outlines the system's functionalities and interactions with external actors. The Use Case Specifications detail the steps for two crucial use cases. The Context Diagram and Level 0 DFD illustrate the system's boundaries, processes, and data flows. Lastly, the ERD captures the relationships between entities in the system.

Together, these models provide a structured and organized representation of the system, laying the foundation for further detailed design and development. The use of these modeling techniques ensures a clear understanding of requirements and sets the stage for effective system implementation.

P7 Discuss, using examples, the suitability of software behavioral design techniques.

Design Phase: Addressing User and Software Requirements:

Addressing User and Software Requirements:

In the design phase, the goal is to translate user and software requirements into a tangible and visually appealing system. This involves creating mock-ups and wireframes to ensure that the user interface aligns with expectations and facilitates efficient collaboration between stakeholders.

Mock-ups and Wireframes:

Mock-up - Customer Login Page:

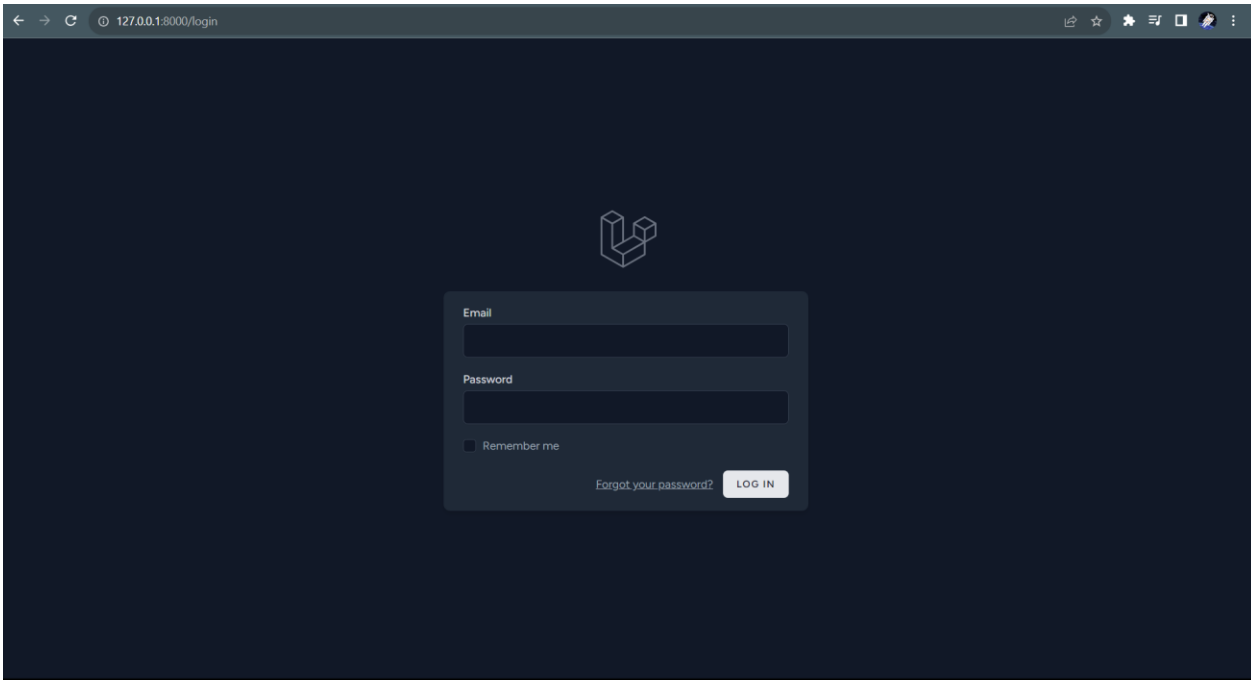
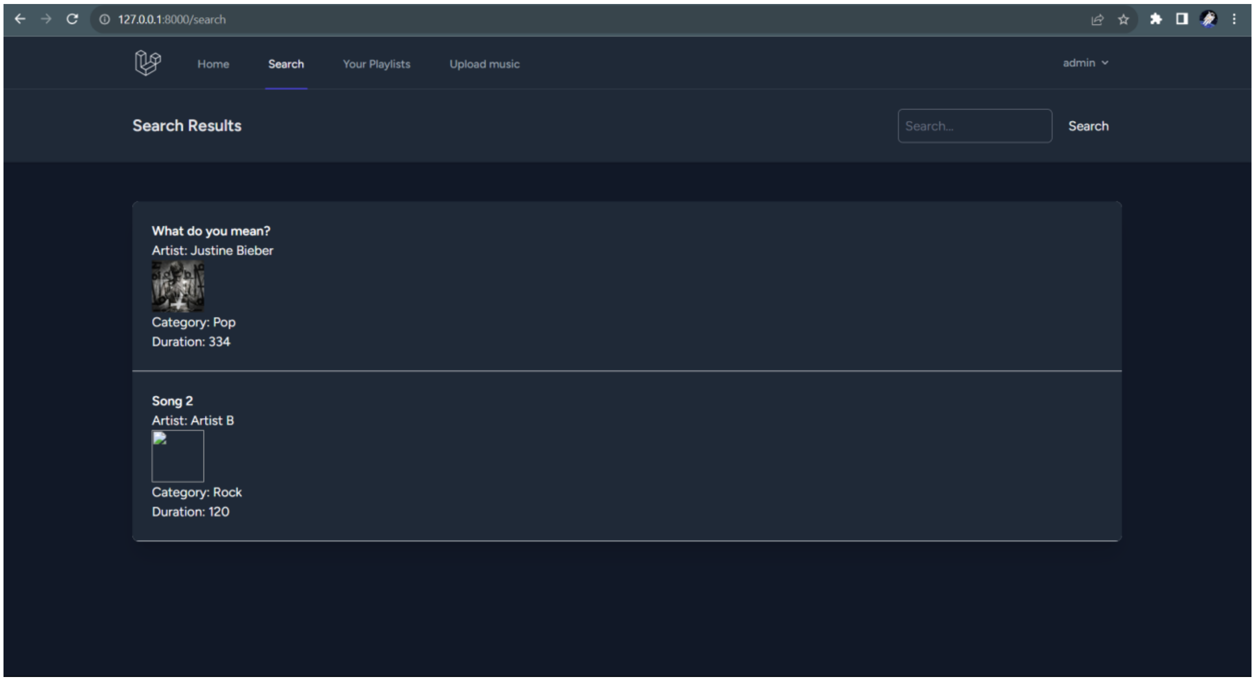


Figure 23: Customer Login Page

Justification: Allows customers to provide feedback on the visual layout, color scheme, and placement of login elements.

Wireframe - Music Search Results Page:



Search Results

What do you mean?
Artist: Justine Bieber
Category: Pop
Duration: 334

Song 2
Artist: Artist B
Category: Rock
Duration: 120

Figure 24: Music Search

Justification: Outlines the basic structure of the search results page, aiding in discussions about the placement of filters, search bar, and result displays.

Mock-up - Customer Registration Page:

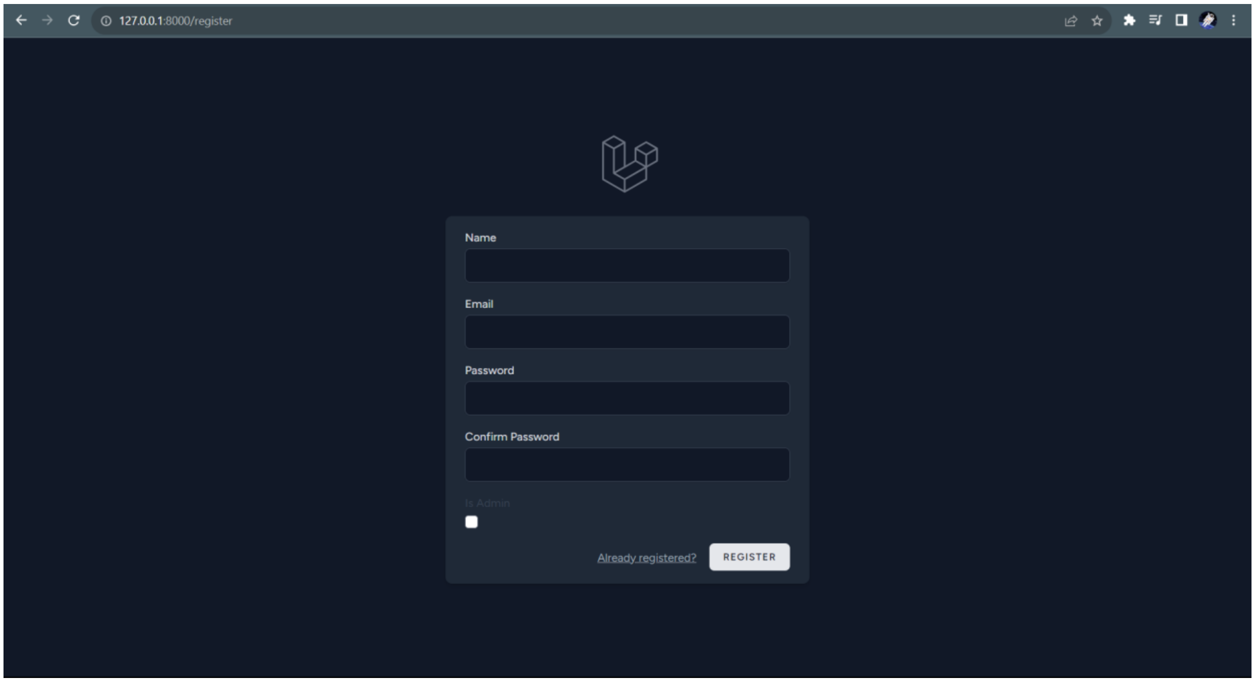
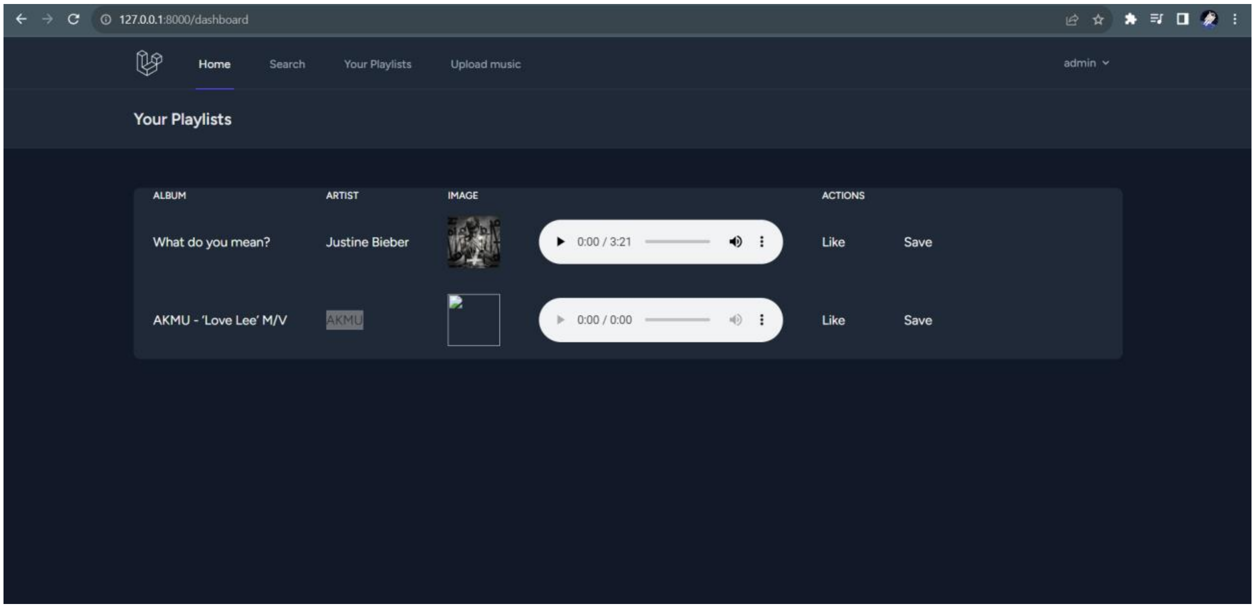


Figure 25: Register

Justification: Visualizes the design of the customer registration interface, allowing stakeholders to provide feedback on the layout, information fields, and user-friendly registration process.

Wireframe - Playlists Management Interface:



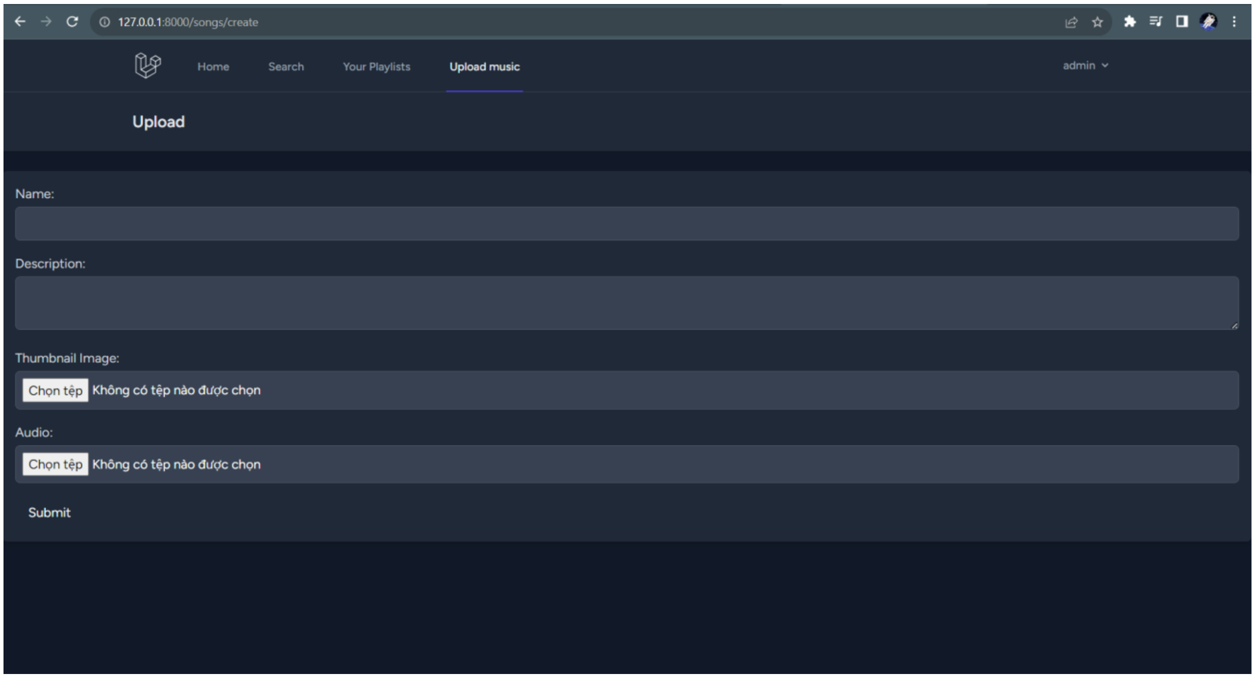
Your Playlists

ALBUM	ARTIST	IMAGE		ACTIONS	
What do you mean?	Justine Bieber		0:00 / 3:21	Like	Save
AKMU - 'Love Lee' M/V	AKMU		0:00 / 0:00	Like	Save

Figure 26: Playlists

Justification: Provides a skeletal representation of the playlists management interface, focusing on the layout and essential interactive elements. This wireframe allows for discussions on playlist creation, editing, and organization.

Mock-up - Music Upload Interface:



Name:

Description:

Thumbnail Image:
Chọn tệp Không có tệp nào được chọn

Audio:
Chọn tệp Không có tệp nào được chọn

Figure 27: Upload

Justification: Illustrates the design of the music upload interface, enabling stakeholders to provide feedback on the flow, file upload options, and metadata entry steps. This mock-up visualizes the process for artists and users to contribute to the music archive.

Different architectures for Tune Source:

Client-Server Architecture:

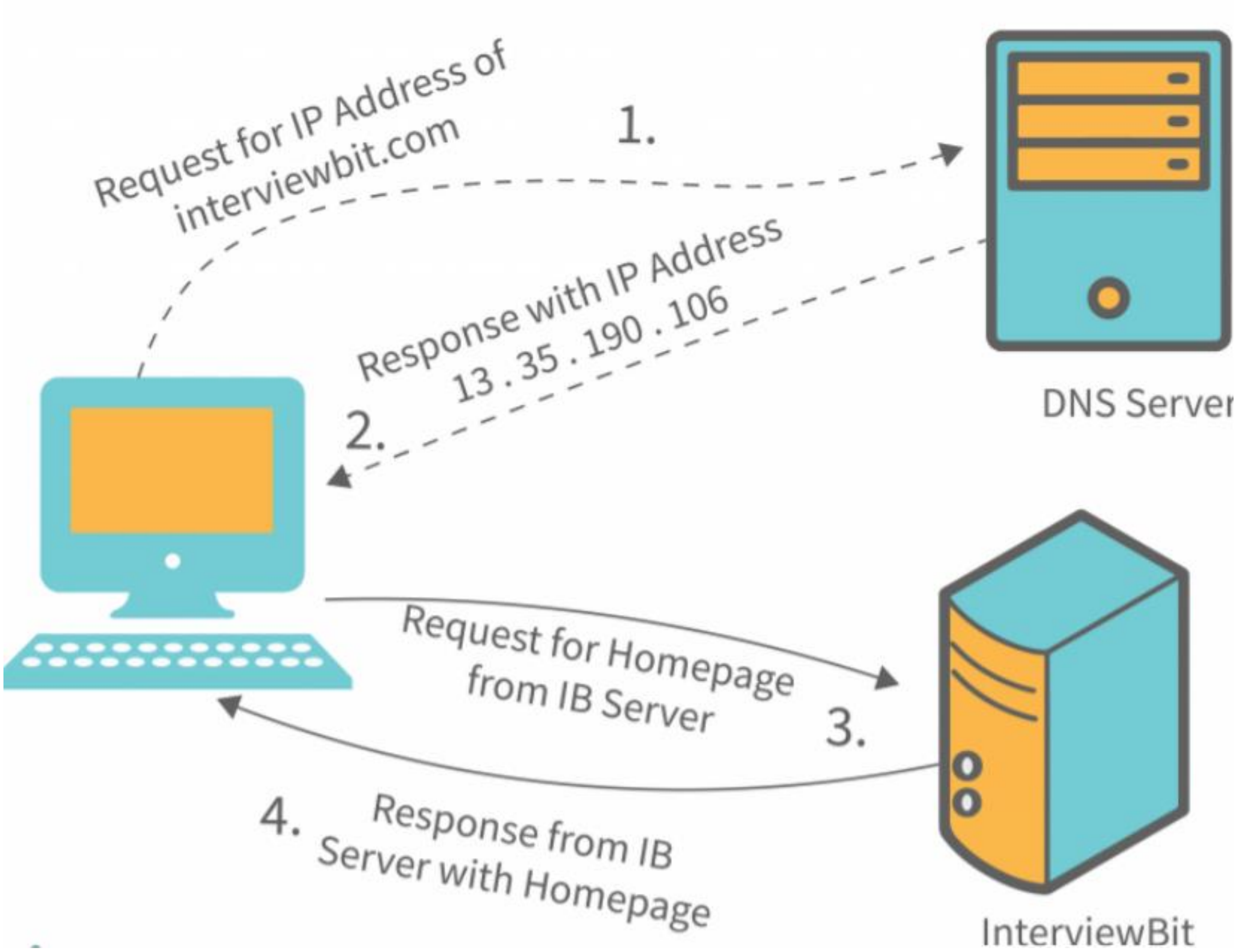

Request for IP Address of interviewbit.com
1.
Response with IP Address 13 . 35 . 190 . 106
2.
DNS Server
Request for Homepage from IB Server
3.
4.
Response from IB Server with Homepage
InterviewBit

Figure 28: Client-Server Architecture

Overview:

In the client-server architecture, the system is divided into two main components: the client, which handles the user interface and user input, and the server, which manages the application logic, database, and business rules.

Advantages:

- **Clear Separation:** Distinct separation simplifies maintenance and updates.
- **Scalability:** Allows for scalable growth by distributing load appropriately.

N-Tier Architecture:

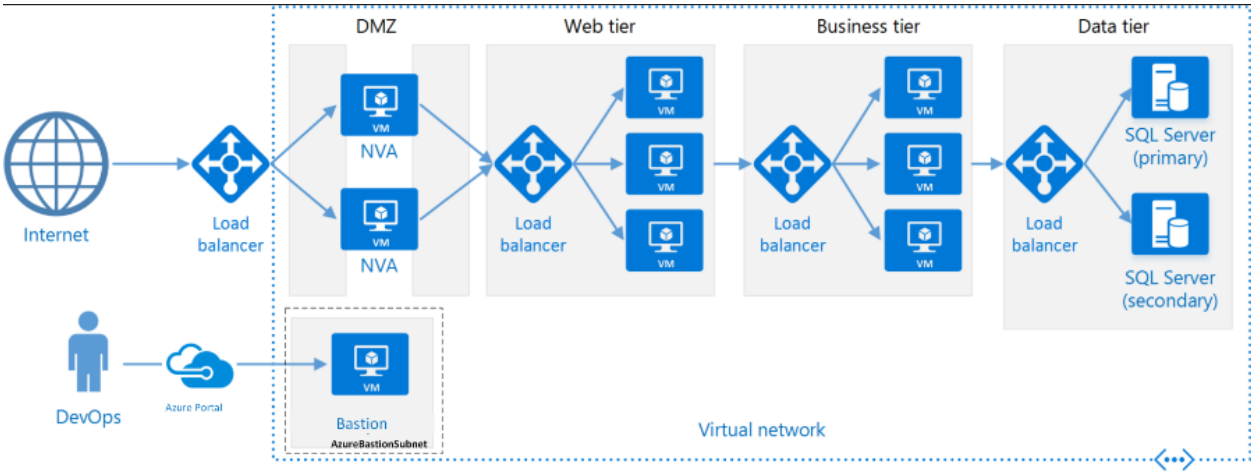


Figure 29: N-tier Architecture

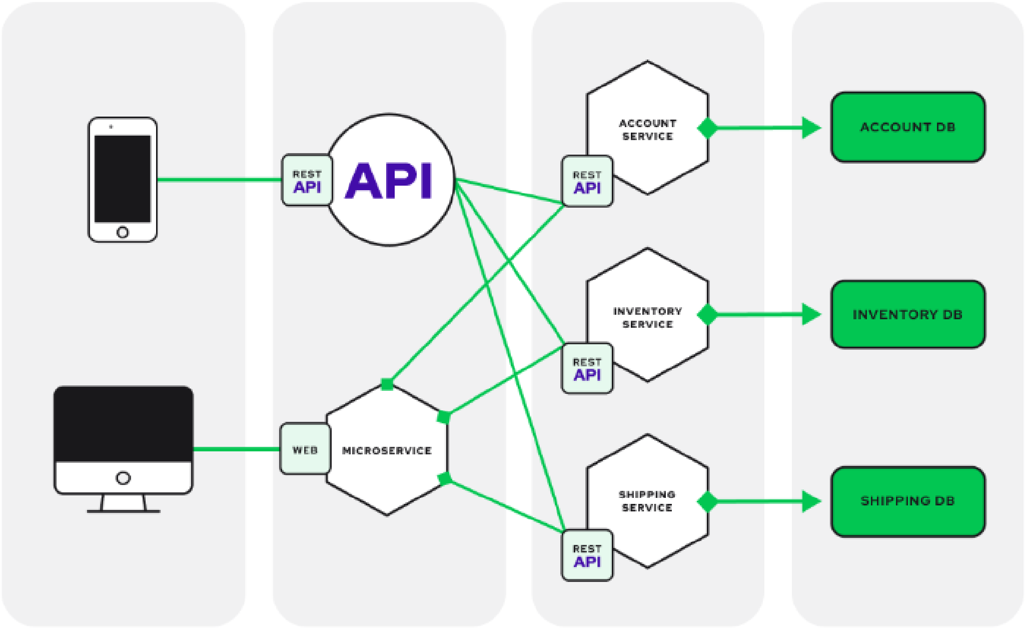
Overview:

N-tier architecture extends the client-server model by introducing additional layers between the client and server, often involving a presentation layer, application layer, and database layer.

Advantages:

- **Modular and Scalable:** Allows for flexibility and scalability by breaking down the system into independent tiers.
- **Improved Security:** Sensitive data can be isolated in a separate layer.

Microservices Architecture:



Overview:

Microservices architecture structures the system as a collection of small, independent services, each handling a specific business functionality.

Advantages:

- **Independent Deployment:** Each microservice can be deployed, updated, and scaled independently.
- **Fault Isolation:** Failure in one microservice doesn't necessarily affect others.

Monolithic Architecture:

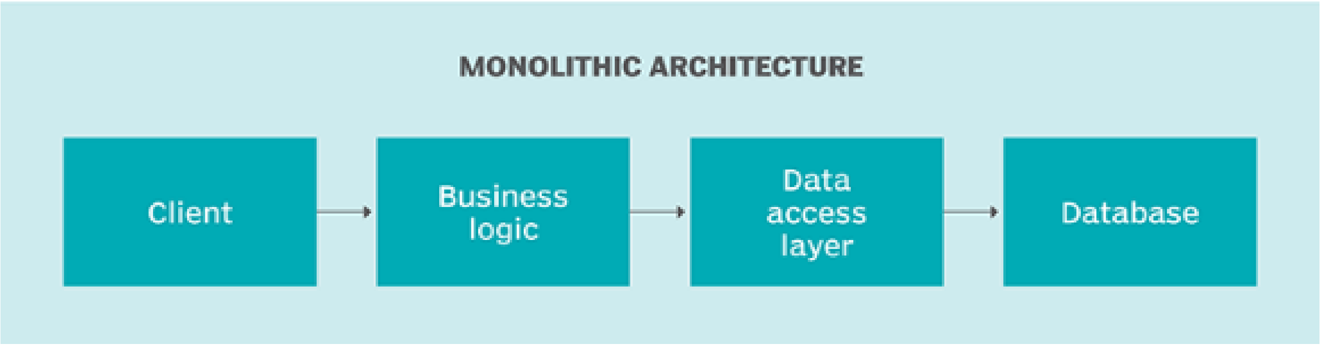


Figure 30: Monolithic Architecture

Overview:

In a monolithic architecture, the entire application is designed as a single, tightly integrated unit.

Advantages:

- **Simplicity:** Easier to develop and deploy as a single unit.
- **Centralized Management:** All components are in one place, simplifying maintenance.

Justification:

Client-Server Architecture is Suitable for Tune Source:

- **Clear Separation:** Given Tune Source's digital music download platform, the client-server architecture provides clear separation between the user interface (client) and the backend services (server).
- **Scalability:** As Tune Source anticipates growth, the client-server model facilitates scalability by allowing the distribution of load appropriately.
- **Simplicity:** For the current scope and scale of the project, a client-server architecture is simpler to implement and maintain compared to more complex architectures like microservices or n-tier.

Technical Solution Stack:

Frontend: React.js

Modular Design: React.js provides a modular and efficient way to build user interfaces, enhancing the overall user experience.

Responsive UI: Enables the creation of a dynamic and responsive user interface, crucial for a seamless music discovery and download platform.

Backend: XAMPP with PHP

Fast and Scalable: XAMPP, with its integration of Apache HTTP Server and PHP, allows for fast and scalable server-side scripting, aligning perfectly with the requirements of a dynamic music platform.

Efficient Request Handling: PHP, in combination with XAMPP, simplifies the handling of HTTP requests, ensuring a smooth and efficient backend operation.

Database: XAMPP MySQL

Flexibility: MySQL, included in the XAMPP stack, offers flexibility in managing music data and customer information, adapting well to the dynamic nature of Tune Source's content.

Efficient Storage and Retrieval: Provides efficient storage and retrieval of data, crucial for managing a vast archive of music tracks and customer records.

Payment Integration: Stripe API

Security: Stripe API ensures secure and seamless payment transactions, meeting the software's financial requirements.

Industry Standard: Widely recognized and used in the industry, offering reliability and a comprehensive set of features for payment processing.

Hosting: AWS (Amazon Web Services)

Reliability: AWS provides reliable and scalable hosting services, crucial for accommodating the potential growth of Tune Source.

Scalability: Ensures the platform can scale with increased demand, offering a robust hosting solution for a dynamic music download platform.

Conclusion

The proposed technical solution stack and system architecture align closely with the outlined requirements and objectives set forth in the introduction for the Tune Source project.

Meeting Business Needs:

The chosen client-server architecture, XAMPP with React.js and PHP, MySQL, Stripe API, and AWS hosting collectively address the business needs identified in the introduction. The technical solution is poised to enhance sales, enable digital music downloads, and tap into new revenue streams through customer subscriptions and gift cards.

Enhancing User Experience:

React.js, with its modular design and ability to create responsive user interfaces, contributes significantly to enhancing the overall user experience. The proposed design phase, including mock-ups and wireframes, ensures that the user interface aligns with customer expectations.

Scalability and Performance:

The client-server architecture, coupled with AWS hosting, emphasizes scalability and reliable performance. This addresses the need to accommodate potential growth in both user interactions and the vast archive of music tracks.

Security and Reliability:

The inclusion of Stripe API for payment integration ensures the security and reliability of financial transactions. Additionally, AWS hosting provides a secure and scalable environment, aligning with the strategic importance of offering digital music downloads to remain competitive.

Flexibility and Adaptability:

The flexibility of the XAMPP stack, especially in local development, allows for iterative testing and quick adjustments. This adaptability is crucial in responding to customer feedback and evolving market demands.

Strategic Positioning:

The technical solution stack positions Tune Source strategically by leveraging existing customer loyalty and responding to their requests for digital music downloads. The system is designed to cross-sell, attracting new customers interested in the unique archive of rare and hard-to-find music.

Mitigating Special Issues:

The solution addresses the special issues mentioned in the introduction, particularly the urgency of bringing the system to the market promptly. The chosen technologies and architecture support a timely development process.

In conclusion, the proposed technical solution stack is tailored to meet Tune Source's business needs, enhance user experience, and position the company strategically in the market. It offers a scalable, secure, and flexible foundation for the successful implementation of digital music downloads, aligning with the company's vision and customer demands.

Reference

1. AltexSoft (2021) *Functional and non-functional requirements: Specification an*, *AltexSoft*. Available at: https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/ (Accessed: 26 November 2023).
2. *Functional vs Non Functional Requirements* (2022) *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/ (Accessed: 26 November 2023).
3. Awati, R. (2023) *What is Joint Application Development (JAD)?: Definition from TechTarget*, *Software Quality*. Available at: https://www.techtarget.com/searchsoftwarequality/definition/JAD (Accessed: 26 November 2023).
4. Gary Austin (2023) *Joint Application Development: Definition, Phases & Methodology - study.com*, *oint Application Development | JAD Definition, Steps & Sessions*. Available at: https://study.com/academy/lesson/joint-application-development-definition-phases-methodology.html (Accessed: 26 November 2023).
5. Interaction Design Foundation - IxDF. (2019, October 17). What is Prototyping?. Interaction Design Foundation - IxDF. https://www.interaction-design.org/literature/topics/prototyping
6. *Software engineering: Prototyping model* (2023) *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/software-engineering-prototyping-model/ (Accessed: 26 November 2023).
7. *What is Use Case Diagram?* (no date) *What is use case diagram?* Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/ (Accessed: 26 November 2023).
8. *UML use case diagram tutorial* (no date) *Lucidchart*. Available at: https://www.lucidchart.com/pages/uml-use-case-diagram (Accessed: 26 November 2023).
9. *What is Use Case Specification?* (no date) *What is use case specification?* Available at: https://www.visual-paradigm.com/guide/use-case/what-is-use-case-specification/ (Accessed: 26 November 2023).
10. (No date) *What is the difference between a use case specification and a use case realization ?* Available at: https://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/ID/648/What-is-the-difference-between-a-use-case-specification-and-a-use-case-realization-.aspx#:~:text=A%20Use%20Case%20Specification%20is,responds%20to%20the%20user%20actions. (Accessed: 26 November 2023).
11. *Context diagram* (2023) *IST Project Management Office*. Available at: https://uwaterloo.ca/ist-project-management-office/tools-and-templates/tools/context-diagram (Accessed: 26 November 2023).
12. (No date) *System context diagram: Description & Examples - study.com*. Available at: https://study.com/academy/lesson/system-context-diagram-description-examples.html (Accessed: 26 November 2023).

13. *Levels in data flow diagrams (DFD)* (2023) *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/levels-in-data-flow-diagrams-dfd/ (Accessed: 26 November 2023).
14. *What is a data flow diagram* (no date) *Lucidchart*. Available at: https://www.lucidchart.com/pages/data-flow-diagram (Accessed: 26 November 2023).
15. Biscobing, J. (2019) *What is entity relationship diagram (ERD)?: Definition from TechTarget*, *Data Management*. Available at: https://www.techtarget.com/searchdatamanagement/definition/entity-relationship-diagram-ERD (Accessed: 26 November 2023).
16. *Entity relationship diagram (ERD)* (no date) *SmartDraw*. Available at: https://www.smartdraw.com/entity-relationship-diagram/ (Accessed: 26 November 2023).
17. *Client-server model* (2022) *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/client-server-model/ (Accessed: 26 November 2023).
18. Contributor, T. (2023) *What is the client-server model? - definition from whatis.com*, *Networking*. Available at: https://www.techtarget.com/searchnetworking/definition/client-server (Accessed: 26 November 2023).
19. Martinekuan (no date) *N-tier architecture style - azure architecture center*, *Azure Architecture Center | Microsoft Learn*. Available at: https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/n-tier (Accessed: 26 November 2023).
20. (No date) *N-tier architecture style: Definition and advantages - indeed*. Available at: https://www.indeed.com/career-advice/career-development/n-tier-architecture (Accessed: 26 November 2023).
21. *What is microservices architecture? | google cloud* (no date) *Google*. Available at: https://cloud.google.com/learn/what-is-microservices-architecture (Accessed: 26 November 2023).
22. Martinekuan (no date) *Microservice architecture style - azure architecture center*, *Azure Architecture Center | Microsoft Learn*. Available at: https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices (Accessed: 26 November 2023).
23. Awati, R. and Wigmore, I. (2022) *What is monolithic architecture in software?*, *WhatIs.com*. Available at: https://www.techtarget.com/whatis/definition/monolithic-architecture (Accessed: 26 November 2023).
24. Atlassian (no date) *Microservices vs. monolithic architecture*, *Atlassian*. Available at: https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith (Accessed: 26 November 2023).
25. Paauwe, M. (no date) *Enterprise architecture for process improvement - DRAGON1*, *Dragon1 Software*. Available at: https://www.dragon1.com/resources/enterprise-architecture/business-process-improvement (Accessed: 26 November 2023).
26. *Business process analysis: Key to efficient business operations* (no date) *Spotfire*. Available at: https://www.spotfire.com/glossary/what-is-business-process-analysis (Accessed: 26 November 2023).
27. *BPR (business process reengineering)* (2022) *Synconics Technologies Pvt. Ltd.* Available at: https://www.synconics.com/erp-business-process-reengineering/ (Accessed: 26 November 2023).