



## Development of Online Music Software by using Agile Model: A case study

Sumanta Chatterjee<sup>1</sup>, Bidisha Baidya<sup>2</sup>, Abhirup Mukerjee<sup>3</sup>, Debendra Dey<sup>4</sup>, Priyanka Chakraborty<sup>5</sup>, Pabitra Kumar Bhunia<sup>6</sup>

<sup>1</sup>Assistant Professor, <sup>23456</sup>UG Student  
<sup>123456</sup>Department of Computer Science and Engineering  
<sup>123456</sup>JIS College of Engineering, Kalyani, India

**Abstract:** Agile is a style of software development technique that foresees the need for flexibility and incorporates a certain amount of pragmatism into the delivery of the finished product. Because it emphasizes on the clean delivery of specific software components rather than the full program, agile software development necessitates a cultural shift in many businesses. The capacity of Agile to support teams in a changing environment while keeping a focus on the effective delivery of business value is one of its advantages. Agile fosters a culture of collaboration, and when teams cooperate and comprehend their particular tasks within the process, efficiency increases throughout the entire organization. Finally, because testing is done continuously throughout development, businesses that employ Agile software development can be sure that the products they are releasing are of superior quality. This offers the chance to adjust as necessary and notify teams of any potential problems. Agile has largely overtaken waterfall as the preferred development technique in most businesses, but it is in danger of being supplanted or even swallowed by DevOps due to its rising popularity.

**Keywords:** Online Music Software, Agile Model.

### I. INTRODUCTION

There are online music stores where customers can buy and download music in addition to streaming services. These shops are becoming more and more well-liked because of how convenient it is to purchase single songs or entire albums without having to leave your house. Online music portals have revolutionized the way we listen to and discover music. With the rise of streaming services like Spotify and Apple Music, people can access millions of songs from all over the world with just a few clicks. These platforms offer personalized recommendations based on user preferences, making it easier than ever to find new artists and genres. In addition to streaming services, there are also online music stores where users can purchase and download music. The convenience of being able to buy individual songs or albums without leaving your home has made these stores increasingly popular. The enormous volume of music offered by online music portals is one of their largest benefits. Millions of songs from throughout the world, including uncommon and difficult-to-find tracks, are available to users. This makes it simpler for music fans to research various genres and find new musicians.

The ease aspect of internet music portals is another advantage. Whether using a computer, phone, or tablet with an internet connection, users may access their favorite songs and playlists. This implies that you no longer need to carry around real CDs or LPs in order to listen to music everywhere you go.

### II. REQUIREMENT OF SYSTEM

The Agile approach, also known as Agile Software Development Life Cycle (SDLC), has a number of benefits. The following are some major benefits of applying the Agile model:

1. Flexibility and adaptability: Agile techniques place a strong emphasis on flexibility. It enables teams to react swiftly to altering demands, priorities, or market circumstances. Agile projects are broken up into shorter iterations, known as sprints, which allow teams to periodically reassess and modify their strategy in response to feedback and changing demands.

2. Client satisfaction: Throughout the development process, Agile places a heavy emphasis on client collaboration. Customers or stakeholders are regularly consulted during feedback sessions to make sure the final product lives up to their expectations and adds value. Agile lowers the risk of delivering a product that doesn't match customer expectations by involving customers frequently and early in the development process.

3. Shorter Iterations: Agile techniques place a strong emphasis on producing usable software quickly. Development teams may quickly and regularly deliver incremental value by segmenting the project into manageable features or user stories. This strategy gives businesses a competitive edge by enabling them to offer a useful product more quickly.

4. Transparency and Visibility: Throughout the development process, Agile encourages transparency and visibility. Team members, stakeholders, and clients can keep up to date on project progress, difficulties, and successes through regular meetings like daily stand-ups and sprint reviews. Transparency improves collaboration and communication while creating trust and agreement among all parties.

5. Early and ongoing risk mitigation: Agile approaches encourage early risk assessment and risk reduction. Delivering functional increments in brief cycles allows for the early identification and mitigation of any problems and hazards. Agile teams are able to quickly modify their plans and methods to manage risks, lowering overall project risk and increasing the likelihood of success.

6. Increased Collaboration and Ownership: Agile approaches foster close engagement amongst team members, including developers, testers, designers, and stakeholders. This results in increased collaboration and ownership. Sharing of expertise, improved problem-solving, and a sense of shared ownership are all encouraged in this collaborative setting. Agile teams self-organize, giving people the freedom to take charge and make decisions, which boosts motivation and productivity.

7. Continuous Improvement: Agile was founded on the idea of ongoing development. Teams undertake retrospectives at the conclusion of each sprint to consider what went well, what may be improved, and to pinpoint steps to improve their processes. Teams can improve gradually over time thanks to this iterative feedback loop, which boosts output, quality, and customer happiness. It's critical to remember that while Agile has many benefits, its success depends on strong teamwork, regular communication, and engaged stakeholders. Although it might not be appropriate for every project or organization, when done effectively, it can have a major positive impact on delivery speed, customer satisfaction, and flexibility.

### III. PROPOSED SYSTEM

- Data Flow Diagram of Level 0:

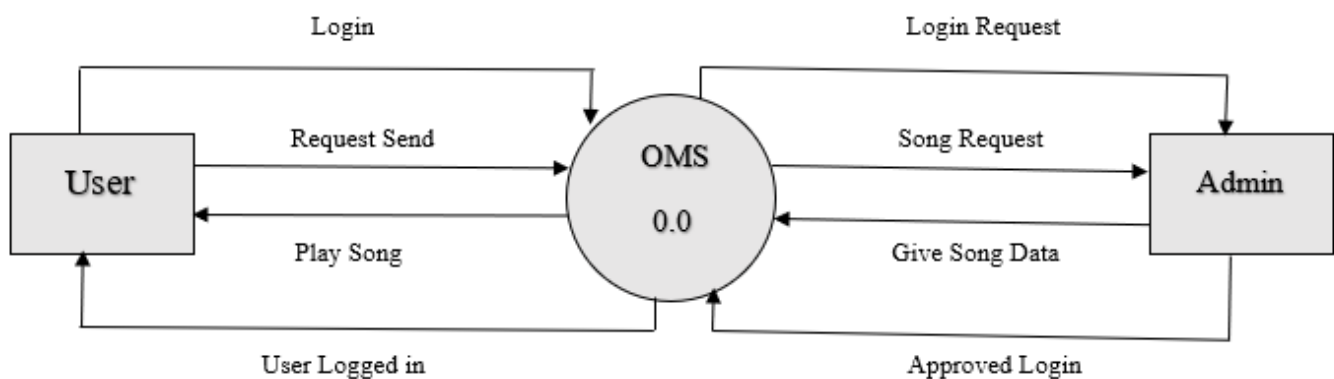


Fig 1: Level 0 DFD of Online Music Software

- Data Flow Diagram of Level 1:

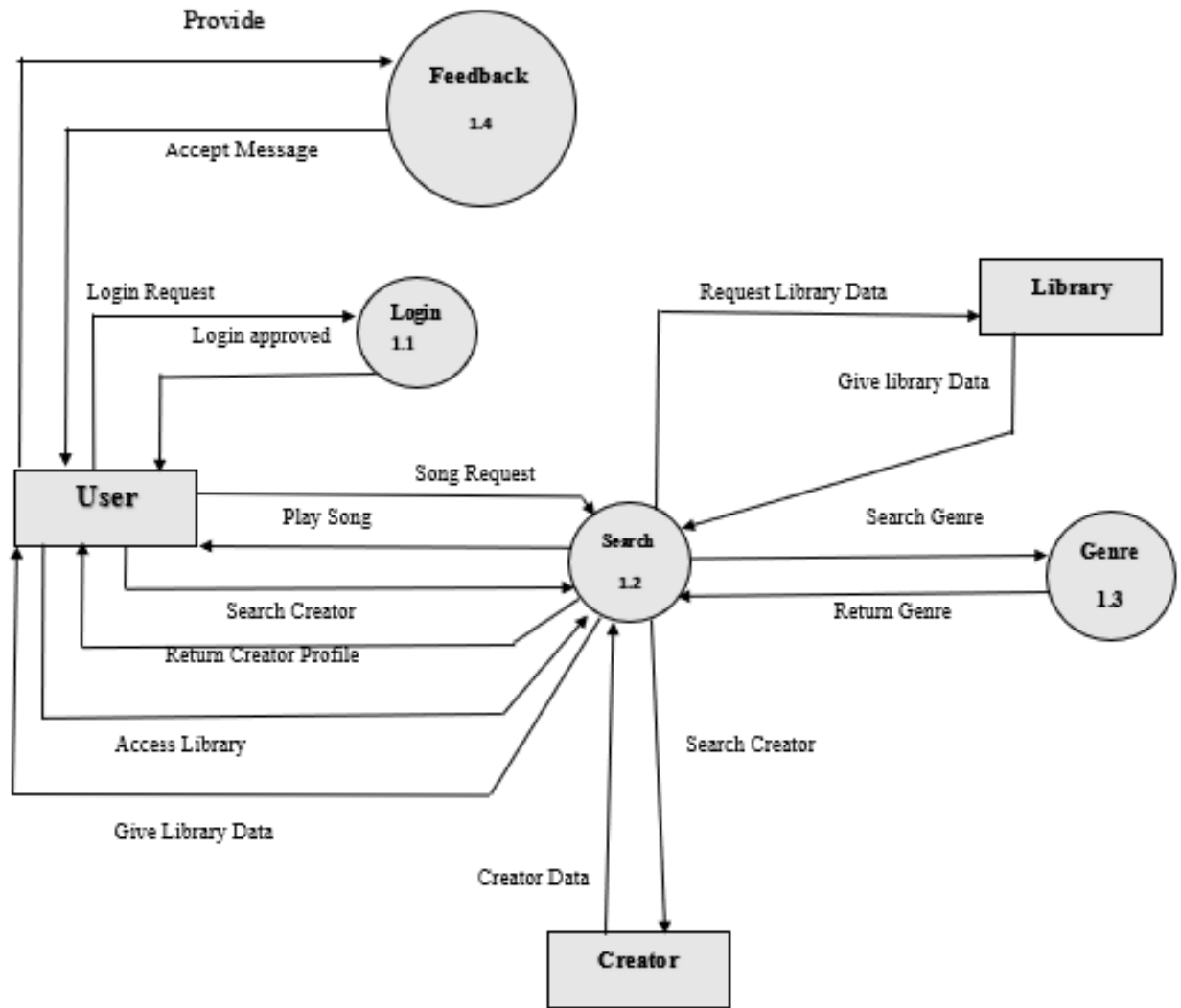


Fig 2: Level 1 DFD of Online Music Software

- Data Flow Diagram of Level 2:

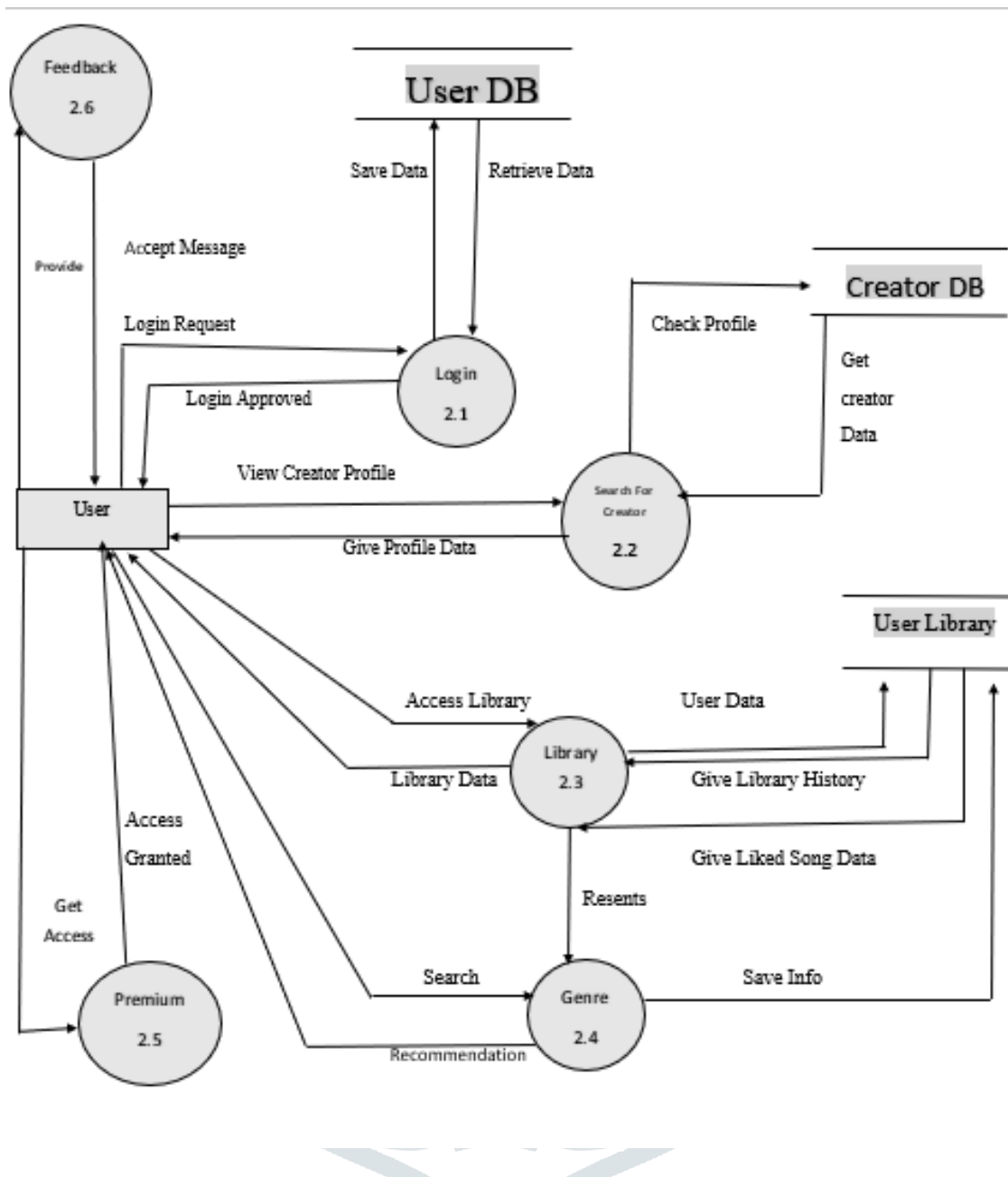


Fig 3: Level 2 DFD of Online Music Software

IV. ENTITY RELATIONSHIP DIAGRAM :

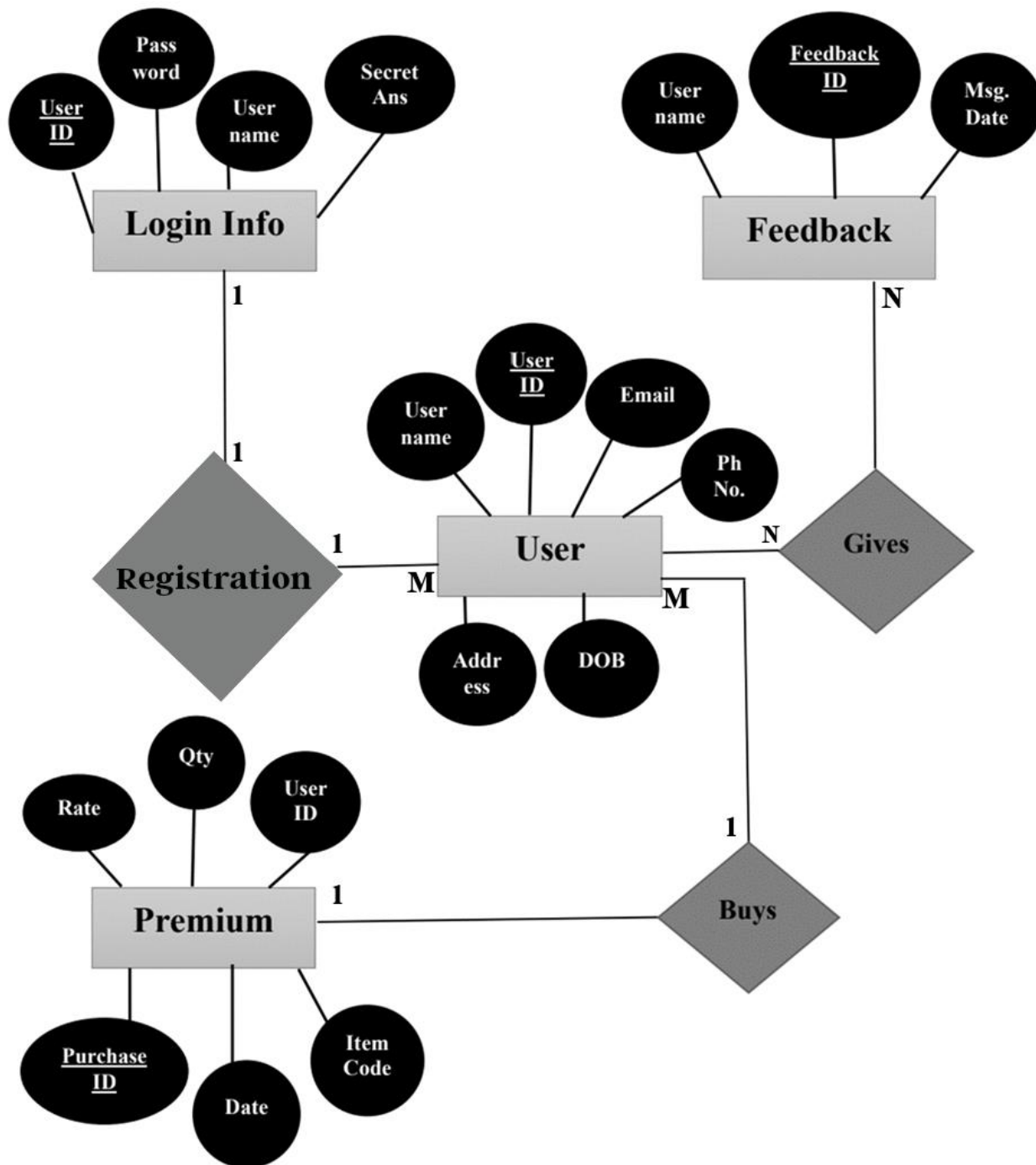


Fig 4: E-R Diagram for Online Music Software

V. USE CASE DIAGRAM:

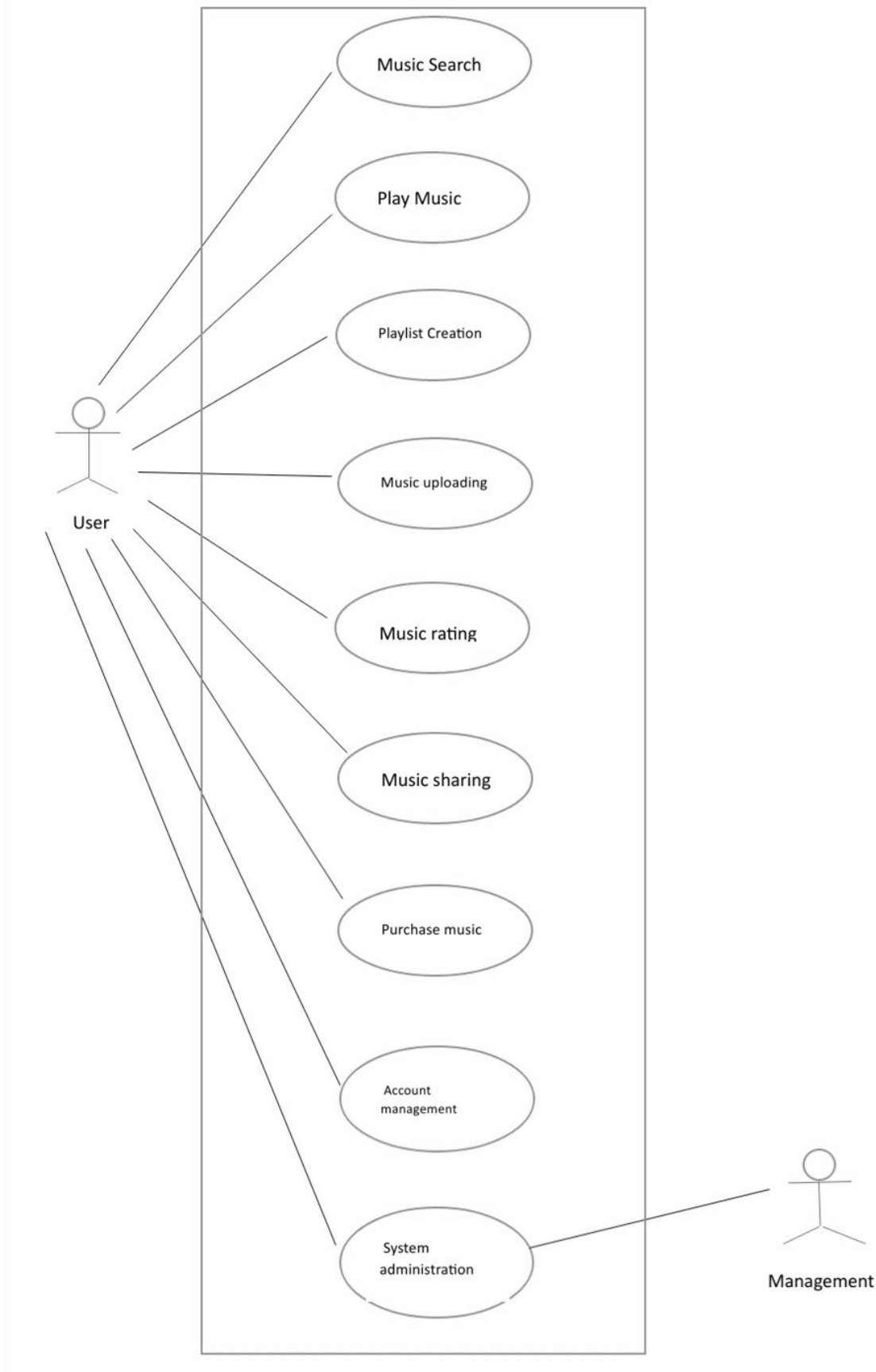


Fig 5: Use Case Diagram for Online Music Software

**EXPLANATION:**

A visual representation of the various interactions between users (actors) and a system is called a use case diagram. A use case diagram would illustrate the various activities and functionalities of an online music program from the viewpoint of the users.

The following is a description of the key components that are frequently shown in a use case diagram for an online music software:

**1. Actors:**

- User: Represents individuals who interact with the online music software. Users are the people who interact with the online music platform. Users can play a variety of roles, including logged-in users, administrators, and artists.

**2. Use Cases:**

- Music Search: Using the online music software, users can look up for certain songs, albums, or artists.
- Play Music: Users can play selected songs or playlists.
- Playlist Creation: By choosing songs from the accessible music library, users can make unique playlists.
- Music Uploading: Music can be uploaded to the platform by artists or authorized users.
- Music Rating: Users can rate songs or entire albums in accordance with their personal tastes.
- Music Sharing: Through a variety of methods, users can share songs, playlists, or their own original music with others.
- Purchase Music: Users can buy individual tracks or entire albums of music for their own use.
- Account Management: Users are capable to change account preferences, including personal data, payment information, and privacy settings.
- Offer Feedback: Users are able to offer feedback or report software-related problems.
- System Administration: Additional management tools for the online music program are available to administrators. These tools include the ability to handle user accounts, copyright disputes, and consumption data.

**3. Interpersonal ties/ Relationship:**

- Association: Connects actors to use cases, demonstrating their participation in certain deeds.
- Generalization/Inheritance: Shows how a specialized actor can acquire traits from a generic actor. A "User" actor in general may be inherited by an "Artist" actor, for instance.
- Include: Shows how two use cases are related to one another by illustrating how one-use case relies on another for some of its capabilities.
- Extend: Illustrates an optional or substitute behaviour that can be applied to a base use case.

**4. System Boundary:**

- Depicts the framework's perimeter, encompassing all actors and use cases, and defining the bounds of the online music platform.

Stakeholders can acquire a high-level knowledge of the capabilities, interactions, and roles of various users of the online music program by using a use case diagram. It provides as a guide for creating software, analyzing requirements, and coordinating with stakeholders.

**VI. COMPARATIVE ANALYSIS**

AGILE MODEL	WATERFALL MODEL	ITERATIVE MODEL	V - MODEL	SPIRAL MODEL
Agile Model is the software development model in which the development and testing process carries on simultaneously. In this model, both development-related processes and testing-related processes are parallel. This model provides the facility for more interaction between the development team, testing team, and end-users.	The waterfall model is highly structured and systematically steps through requirements gathering, analysis, SRS document preparation, design, coding, and testing in a planned manner. These phases of the Waterfall model follow a sequential order.	The iterative process model is a software development life cycle (SDLC) approach in which the initial development work is conducted based on initial requirements that are clearly defined, and subsequent features are added to this base software product through iterations until the final system is completed.	V-Model is the software development model in which testing takes place once the development process is fully complete or almost complete. The V-Model development and testing process are kept quite separate.	<b>Spiral model</b> is one of the most important Software Development Life Cycle models, which provides support for <b>Risk Handling</b> . In its diagrammatic representation, it looks like a spiral with many loops.
The agile model allows to change the requirements after the development process	The waterfall model is rigid, it does not allow to change requirements after the	Comparatively less flexible.	This model is less flexible and more rigid compared to the agile model.	This one is flexible but agile model is more flexible.

starts, so it is more flexible.	development process starts.			
It is suitable for small or large projects where the work needs to be completed iteratively.	This model is simple to use and understand but not suitable for developing large projects using the Waterfall model.	It is suitable to use in case a required change affects the large numbers of components as major changes are needed by relevant features.	It is suitable for large projects where the work needs to be completed sequentially.	The Spiral model is suitable for those projects that are prone to various kinds of risks that are difficult to anticipate at the beginning of the project.
Customer interaction is very high. After each iteration, an incremental version is deployed to the customer.	Customer interaction is very less. The product is delivered to the customer after the overall development is completed.	Less customer interaction compared to agile model.	The development team and testing team do not interact much with end users.	This has better customer interaction also but agile is comparatively better.

## VII. CONCLUSION & FUTURE SCOPE

The music industry has been significantly impacted by online music portals. They have ruined conventional distribution methods and made record companies change how they operate. With millions of customers worldwide, streaming services like Spotify and Apple Music have established themselves as key participants in the market. The preference for singles over albums is one of the most significant developments brought about by internet music portals. Many listeners are no longer interested in purchasing complete albums now that you can buy individual singles. Because of this, many artists and record companies now prioritize the release of singles and EPs above full-length albums. Online music portals have transformed the way we listen to and discover music. They offer unparalleled convenience and access to a vast library of songs and artists from all over the world. While there are certainly challenges facing the industry, the potential for innovation and growth is enormous. As technology continues to evolve, we can expect online music portals to become even more sophisticated and user-friendly. Whether you're a casual listener or a die-hard music fan, these platforms offer something for everyone.

## VIII. REFERENCES

- [1] . "AGILE SOFTWARE DEVELOPMENT METHODS: REVIEW AND ANALYSIS", ABRAHAMSSON P, SALO O, RONKAINEN J, WARSTA JARXIV PREPRINT ARXIV:1709.08439. 2017 SEP 25.
- [2] "MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT", BECK, K., BEEDLE, M., VAN BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A., JEFFRIES, R. AND KERN, J., 2001
- [3] "Agile software development: The business of innovation.", Highsmith, Jim, and Alistair Cockburn *Computer* 34.9 (2001): 120-127.
- [4] 'Agile software development'. Cohen D, Lindvall M, Costa P, DACS SOAR Report. 2003 Jan 31;11:2003.
- [5] "Agile software development: principles, patterns, and practices", Martin R, Prentice Hall PTR; 2003 Sep 1.
- [6] ". Agile software development. Software: Practice and Experience", Greer D, Hamon Y, 2011 Aug;41(9):943-4.
- [7] "Agile software development.", Koch, Alan, Artech, 2004.
- [8] "An introduction to agile software development", Szalvay V, Danube technologies. 2004 Nov;3.
- [9] "Music software in the technology integrated music education", Nart S, Turkish Online Journal of Educational Technology-TOJET. 2016 Apr;15(2):78-84.
- [10] "The system development life cycle (sdlc)", Radack S., National Institute of Standards and Technology; 2009 Apr 29.