|  |
| --- |
| **Project Document for Software Project Lab - 3** |
| **Encryption Based**  **Audio File Distribution System**  **With**  **Compatible Music Player** |
| **Institute of Information Technology, University of Dhaka** |



Project Document on

**Encryption based Audio File Distribution System with Compatible Music Player**

Software Project Lab - III

SE 801

Submitted to:

BSSE​ ​4​th year​ ​Program​ ​Committee

IIT, University of Dhaka

Supervised by

Dr. Md. Shariful Islam

Professor

IIT, University of Dhaka

Submitted by

Md. Ruhul Amin Rahat (BSSE 0616)

Submission Date

14th December, 2017

### Document Authentication

This project document has been approved by the following persons:

………………………………………… …………………………………….........

**Approved by:**  **Prepared by:**

Dr. Md. Shariful Islam Md. Ruhul Amin Rahat

Professor BSSE 0616

IIT, University of Dhaka IIT, University of Dhaka

**Letter of Transmittal**

BSSE​ ​4​th​ year​ ​Program​ ​Committee

Institute of Information Technology,

University of Dhaka

December 14, 2017

Sir,

I have prepared the enclosed report of **Encryption based Audio File Distribution System With Compatible Music Player** for your approval. This report details the requirements I gathered for the project.

The primary purpose of this document is creating SRS, design, testing report for the project I am doing for my *Software Project Lab-III*. This report includes the details of each steps we followed to collect the requirements.

Sincerely Yours,

*Md. Ruhul Amin Rahat (BSSE0616)*

*4nd Year, 8th Semester, 6th Batch*

*Institute of Information Technology*

*University of Dhaka*

Session: 2013-14

Enclosure: Software Requirement Specification, Design and Testing Report

**Executive Summary**

The purpose of **Encryption based Audio File Distribution System With Compatible Music Player**, which contains an android app, is to provide a distribution system of audio music files for musicians with the help of a payment gateway and to provide a music player for customers, compatible with the distribution system. The system focuses on solving the piracy problem in audio music industry and on providing revenue to musicians. This document provides the Software Requirements Specification, Architectural Design, Component Design, User Interface Design, Test Cases to develop the system.

**Acknowledgement**

I am expressing my heartiest gratitude to Almighty ALLAH to complete the designated Project Report in time and without hassles.

I would like to thank my project supervisor Dr. Md. Shariful Islam and course instructor Amit Seal Ami and Nadia Nahar for their valuable supervision, advice, instruction and time throughout the project.

# 

**Table of Contents**

[Chapter 1: Introduction 1](#_1fob9te)

[1.1 Purpose 1](#_2et92p0)

[1.2 Intended Audience 1](#_tyjcwt)

[Chapter 2: Inception 2](#_3dy6vkm)

[2.1 Introduction 2](#_4d34og8)

[2.1.1 List of Stakeholders 2](#_2s8eyo1)

[2.1.2 Recognizing multiple viewpoint 3](#_17dp8vu)

[2.1.3 Working towards collaboration 4](#_3rdcrjn)

[2.2 Conclusion 5](#_26in1rg)

[Chapter 3: Elicitation 6](#_35nkun2)

[3.1 Introduction 6](#_1ksv4uv)

[3.2 Eliciting requirements 6](#_44sinio)

[Collaborative Requirements Gathering 6](#_2jxsxqh)

[Quality Function Deployment (QFD) 7](#_z337ya)

[Usage Scenario 8](#_3j2qqm3)

[Elicitation Work Product 9](#_1y810tw)

[3.3 Conclusion 9](#_2xcytpi)

[Chapter 4: Scenario Based Modeling 10](#_3whwml4)

[4.1 Introduction 10](#_2bn6wsx)

[4.2 Use case Scenario 10](#_qsh70q)

[4.3 Use Case Diagrams 10](#_3as4poj)

[Level-0 use case: 11](#_1pxezwc)

[Level-1 use case: 12](#_147n2zr)

[Subsystems of Level-1.1 Use Case 13](#_32hioqz)

[Subsystems of Level-1.2 Use Case 14](#_vx1227)

[Subsystems of Level-1.3 Use Case 15](#_4f1mdlm)

[Subsystems of Level-1.4 Use Case 16](#_28h4qwu)

[Subsystems of Level-1.5 Use Case 17](#_1mrcu09)

[Subsystems of Level-1.6 Use Case 18](#_111kx3o)

[Subsystems of Level-1.7 Use Case 19](#_2zbgiuw)

[4.4 Activity & Swimlane Diagrams 20](#_3cqmetx)

[Chapter 5: Data Model 26](#_3hv69ve)

[5.1 Data Modeling Concept 26](#_1x0gk37)

[5.2 Data Objects 26](#_4h042r0)

[5.2.1 Identify Data Objects 26](#_2w5ecyt)

[5.2.3 Entity Relationship Diagram 28](#_1baon6m)

[Chapter 6: Class Based Model 29](#_2afmg28)

[6.1 Class Modeling Concept 29](#_pkwqa1)

[6.1.1 Identifying Analysis Class 29](#_39kk8xu)

[6.1.2 Class Responsibility Collaboration (CRC) Diagram 33](#_1opuj5n)

[Chapter 7: Behavioral Model 34](#_1302m92)

[7.1 State Transition Diagram 34](#_3mzq4wv)

[State transition diagrams for all classes: 34](#_2250f4o)

[7.2 Sequence Diagram 37](#_3ep43zb)

[Chapter 8: Software​ ​Design​ ​Architecture 38](#_4du1wux)

[8.1​ ​Introduction 38](#_2szc72q)

[8.2​ ​Representing​ ​the​ ​system​ ​in​ ​context 38](#_184mhaj)

[8.3​ ​Define​ ​archetypes: 39](#_279ka65)

[8.4​ ​Refining​ ​archetypes​ ​into​ ​components: 39](#_meukdy)

[8.5 Describing instantiation of the system 39](#_45jfvxd)

Chapter 9: Component Level Design 42

Chapter 10: User Interface Design 52

10.1 Introduction 52

10.2 Event Transition 53

Chapter 11: Test Plan and Test Cases 67

[11.1 Test Plan Identifier 67](#_1d96cc0)

[11.2 Introduction 67](#_3x8tuzt)

[11.2.1 Summary of items and features to be tested 67](#_2ce457m)

[11.2.2 Requirement and history of items 67](#_rjefff)

[11.2.3 High level description of testing goals 67](#_3bj1y38)

[11.2.4 Reference Document 68](#_1qoc8b1)

[11.3 Test items 68](#_4anzqyu)

[11.4 Features to be tested 68](#_2pta16n)

[11.5 Features not to be tested 68](#_14ykbeg)

11.6 Approach 68

11.7 Item pass/fail criteria 69

11.8 Test Deliverables 69

11.9 Test Cases 69

[Chapter 12 : User Manual 76](#_3oy7u29)

[Chapter 13 : Conclusion 82](#_3oy7u29)

**List Of Figures**

Figure 4.1: Level-0 Use case diagram 11

Figure 4.2: Level-1 Use case diagram 12

Figure 4.3: Level-1.1 Use Case Diagram 13

Figure 4.4: Subsystems of Level-1.2 Use Case Diagram 14

Figure 4.5: Subsystems of Level-1.3 Use Case Diagram 15

Figure 4.6: Subsystems of Level-1.4 Use Case Diagram 16

Figure 4.7: Subsystems of Level-1.5 Use Case Diagram 17

Figure 4.8: Subsystems of Level-1.6 Use Case Diagram 18

Figure 4.9: Subsystems of Level-1.7 Use Case Diagram 19

Figure 4.10: Activity Diagram for use case 1.1.1 20

Figure 4.11: Swimlane Diagram for use case 1.1.1 21

Figure 4.12: Activity Diagram for use case 1.1.2 22

Figure 4.13: Swimlane Diagram for use case 1.1.2 22

Figure 4.14: Activity Diagram for use case 1.1.3 23

Figure 4.15: Swimlane Diagram for use case 1.1.3 23

Figure 4.16: Activity Diagram for use case 1.1.4 23

4.17: Activity Diagram for use case 1.1.5 23

Figure 4.18: Activity Diagram and Swimlane Diagram for use case 1.1.6 24

Figure 4.19: Activity Diagram and Swimlane Diagram for use case 1.1.7 25

Figure 5.1: ER Diagram for data based modeling 28

Figure 6.1: CRC Diagram 34

Figure 7.1: State Transition Diagram for Customer Class 35

Figure 7.2: State Transition Diagram for Artist Class 35

Figure 7.3: State Transition Diagram for Artist Profile Class 36

Figure 7.4: State Transition Diagram for File Class 36

Figure 7.5: State Transition Diagram for Post Class 36

Figure 7.6: State Transition Diagram for Music Player Class 37

Figure 7.7: Sequence Diagram 38

Figure 8.1: Architectural Context Diagram 39

Figure 8.2: Refining ‘Authentication’ archetype into components and classes 40

Figure 8.3: Refining ‘File Upload and Revenue info’ archetype into components and classes 41

Figure 8.4: Refining ‘Find artists and posts’ archetype into components and classes 41

Figure 8.5: Refining ‘Find purchase and music player’ archetype into

components and classes 41

Figure 9.1 - 9.4 : Class Components 43 - 46

Figure 9.5 : Specifying message details 47

Figure 9.10 : Elaborate Deployment 51

Figure 10.1 - 10.12 : User Interface Design 55 - 66

**Acronyms**

|  |  |
| --- | --- |
| **Acronyms** | **Definition** |
| SRS | Software Requirement Specification |
| EAFDSCMP | Encryption Based Audio File Distribution System With Compatible Music Player |
| QFD | Quality Function Deployment |
| ER | Entity Relationship |
| CRC | Class Responsibility Collaboration |
| DAL | Data Access Layer |

### Chapter 1: Introduction

In this chapter, the purpose of the SRS and its intended audience will be described.

## 1.1 Purpose

This document is the Software Requirement Specification (SRS) for the **Encryption based Audio File Distribution System With Compatible Music Player.** It contains functional, non-functional and support requirements and establishes a requirements baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered, and organized by topic. The SRS serves as official means of communicating user requirements to the developer and provides a common reference point for both the developer team and stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

## 1.2 Intended Audience

This SRS is intended for several audiences including the customers as well as the project managers, designers, developers, and testers.

* The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer.
* The project managers of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the developing team is on track during development of the system.
* The designers will use this SRS as a basis for creating the system’s design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer’s needs.
* The developers will use this SRS as a basis for developing the system’s functionality. The developers will link the requirements defined in this SRS to the software they create.
* The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the app are completed, the testers will run their tests to ensure that the app fulfills the requirements documented in this SRS.

### Chapter 2: Inception

In this chapter, the Inception part of the SRS will be discussed briefly.

## 2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how does a software project get started and what is the scope and nature of the problem to be solved. The goal of the inception phase is to identify concurrence needs and conflict requirements among the stakeholders of a software project. At project inception, we establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired, and the effectiveness of preliminary communication and collaborations between the other stakeholders and the software team.

To establish the groundwork I have worked with the following factors related to the inception phases:

* List of stakeholders
* Recognizing multiple viewpoints
* Working towards collaboration
* Requirements questionnaire

## 2.1.1 List of Stakeholders

Stakeholder refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that may be affected by its installation. At inception, a list of people who will contribute input as requirements are elicited (Chapter 3) is created.To identify the stakeholders I consulted with some users and asked them following questions:

* Who is paying for the project?
* Who will be using the project outcomes?
* Who gets to make the decisions about the project (if this is different from the money source)?
* Who has resources I need to get the project done?
* Whose work will my project affect? (During the project and also once the project is completed).

Concluding thoughts on Stakeholders, I identified the following stakeholders :

**Artists:** Persons who are musicians or related to music e.g, singers, songwriters, lyricists are under this classification. Artists are the most important actor of the system. They will use the system by a mobile app and upload songs and will earn revenue for each songs.

**Customers:** People who are loves music or listens music often are under this classification. They are very important actor of the system. They will use the system by a mobile app and purchase and download songs and provide revenue for artists.

**Payment Gateway:** The entity which provide service for easy payment through mobile phone are under this classification. For this project, the payment gateway is phone operator. They are also very important actor of the system. They will consume the song purchase information of customers and provide revenue for artists.

**Developers:** Developers are stakeholder because they develop this system and work for further development. If occurs any system interruption, they will find the problem and try to solve it. For this project, there is only one developer involved.

## 2.1.2 Recognizing multiple viewpoint

Different stakeholders achieve different benefits. Consequently, each of them has a different view of the system. So we have to recognize the requirements from multiple points of view, as well as multiple views of requirements. Assumptions are given below:

**Artist's viewpoint:**

* Provide revenue for each songs
* Friendly User interface
* Efficient mobile apps

**Customer’s viewpoint:**

* User friendly and efficient system
* Error free system
* Availability of expected requirements within the budget
* No harmful effects of their existing technology

**Payment Gateway’s viewpoint:**

* Efficient system
* Error free system
* Provide easy API

**Developer’s viewpoint:**

* Easy to develop
* No ambiguous requirement

## 2.1.3 Working towards collaboration

Every stakeholder has their own requirements. I followed following steps to merge these requirements.

* identify the common and conflicting requirements
* Categorize the requirements
* Take priority points for each requirements from stakeholders and on the basis of this voting prioritize the requirements
* Make final decision about the requirements

Common requirements:

* Fully error free system
* User friendly and efficient system
* Easy to operate
* Authentication
* Database containing detailed information

Conflicting requirements:

I found some requirements conflicting each other. I had to trade-off between the requirements.

* Limited budget
* Availability of all requirements within the budget
* No ambiguous requirement
* Easy access
* Strong authentication and high security
* No harmful effects on existing technology

Final requirements:

We finalized following requirements for the system by categorizing and prioritizing the requirements.

* Error free system (Maximum 5% error may be considerable)
* Restrict access to functionality of the system based upon user roles
* Efficient system
* User friendly and efficient system

## 2.2 Conclusion

Inception stage assisted me to understand the project scope, to identify the stakeholders who will be benefited if the app is developed, defining the core functions of it and establishing the preliminary communication with our stakeholders. Based on the discussion of inception phase I will discuss about the elicitation phase of this report which combines elements of problem solving, elaboration, negotiation and specification in our next chapter.

### 

### Chapter 3: Elicitation

The purpose of this chapter is to specify the elicitation part.

## 3.1 Introduction

Requirements elicitation is a part of requirement engineering that is the practice of gathering requirements from the users, customers, and other stakeholders. I have faced many problems like understanding the problems, problems of making questions for the stakeholders, problems of less communication with the stakeholders for time limitation, problems of volatility. Though it is not too easy to gather requirements within a very short time, I have surpassed these problems in an organized and systematic manner.

## 3.2 Eliciting requirements

We have seen Question and Answer (Q&A) approach in the previous chapter where the inception phase of requirement engineering has been described. The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. We have finished the following tasks for eliciting requirements-

* Collaborative Requirements Gathering
* Quality Function Deployment
* Usage Scenarios
* Elicitation work products

##### 3.2.1 Collaborative Requirements Gathering

Many different approaches to collaborative requirements gathering have been proposed. Each makes use of a slightly different scenario. I completed following steps to do it.

The meetings were conducted with the Customers. They were questioned about their requirements and expectations from the project.

They were asked about the problems they are facing with the current system. At last I selected my final requirement list from the meetings.

##### 3.2.2 Quality Function Deployment (QFD)

Quality Function Deployment (QFD) is a technique that translates the needs of the people into technical requirements. With respect to my project the following requirements are identified by a QFD.

**Normal requirements**

Normal requirements consist of objectives and goals that are stated during the meeting with the customers. Normal requirements of our project are-

* Allow valid users to login and logout
* Fully error free activity
* Efficient and error free file purchase
* Help feature to explain, what they are looking for
* Graphical menu
* Efficient and user friendly
* The user interface of the both the mobile app would be easy and attractive
* Security issue
* Describe the complexity of existing system
* Proper use of resource
* A product reference manual describing how to install, setup, and run the application will be provided

**Expected requirements**

These requirements are implicit to the system and may be so fundamental that the customer does not explicitly state them .Their absence will be a cause for dissatisfaction.

* Maintain a databases of all items to keep all the information
* Security issue will be ensured
* The system would be allow the artist to login based on assigned artist ID and password
* The system would automatically perform the encryption process
* If any user forgets password, show the recovery procedure
* Operational correction
* The user interface of the app shall be easy to use and shall make use of drop-down boxes, radio buttons, and other selectable fields wherever possible instead of fields that require the user to type in data

**Exciting requirements**

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present.

* Good graphical presentation
* The user interface should provide appropriate error message for invalid input
* Social networking features will be updated automatically and will be fast

##### 3.2.3 Usage Scenario

The system will be used by artists (or musicians), customers and guest users. An artist him/herself is both a customer and an artist.

Before signing up, an user can enter the system as a guest user. When a customer first sign up to the system, s/he will also provide his/her phone number (for payment gateway) and password. After signing up the system as a customer, an artist will request to verify him/her as an artist. Once an artist is verified to the system, a profile of the artist is created and an artist ID will be provided to him/her. Customers have option to follow and unfollow artists and “purchase and download” the published songs from the profile view. The artist will select a file from device, input the file’s name and the system will generate a file ID. The system will embed the file ID and artist ID to that file and encrypt that file. Then artist will upload

* the encrypted file
* a preview raw file for that encrypted file and
* a text

The encrypted file, the preview file and the text, as a whole, is considered as “Post” and it will be shown to artist’s profile. Artist’s uploaded file numbers and his/her follower numbers will also be shown in his/her profile.

Artist will also view the numbers of his/her purchased files by customers and his/her revenues for each songs.

A customer will perform the following functions:

**Purchase File:** S/he can purchase any files produced by any artists for upto 3 devices.

**Play Music:** A music player tab, which will deliver the functionality of basic music player for the both the purchased files and ordinary .mp3 files.

**Discover songs and artists:** Customer will view top artist list according to artists’ downloaded song numbers, top downloaded file list and most recent uploaded songs list and s/he can “purchase and download” song from this lists.

**View Feed:** A Customer will view the posts (based on time) from his followed artists on Feed tab. Customer can listen the preview of songs online for a limited amount of time and have option to “purchase and download” songs from here.

**Search Songs and Artists:** Customers will randomly search songs and artists and “purchase and download” songs from that search list result.

A server database will provide data for payment gateway about every artist’s revenue and every customer’s usage. A guest user can perform “Discover songs and artists” and “search songs and artists” functionality but will not be able to download or purchase song. Guest user will be able to play ordinary .mp3 files from his/her device on music player.

##### 3.2.4 Elicitation Work Product

The output of the elicitation task can vary depending on size of the system or product to be built. My elicitation work product includes:

* Making a statement of our requirements for the system.
* Making a bounded statement of scope for our system.
* Making a list of customers, artists, and other stakeholders who participated in the requirements elicitation.
* Making a list of requirements that are organized by function and domain constraints that apply to each.
* A set of usage scenarios that provide insight into the use of the system.
* Description of the system’s technical environment.

## 3.3 Conclusion

Elicitation phase helped me to understand about the problems of our scopes of the system. This phase also helped to identify the requirements, negotiate different approaches and specify a preliminary set of solution requirements in an atmosphere that is conducive to the accomplishment of the goal.

### Chapter 4: Scenario Based Modeling

This chapter describes the scenario based model for Encryption based Audio File Distribution System With Compatible Music Player.

## 4.1 Introduction

In this model the system is described from the user’s point of view. As this is the first model, it serves as input for creation of other modeling elements.

## 4.2 Use case Scenario

A use case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a use case tells a stylized story about how an end user interacts with the system under a specific set of circumstances.

A use case diagram simply describes a story using corresponding actors, who perform important role in the story and makes the story understandable for the users. The first step in writing a use case is to define that set of “actors” that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using system.

**Primary Actor:** Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.The primary actors of this system are artist, customer and guest user

**Secondary Actor:** Secondary actors support the system so that primary actors can do their work. They either produce or consume information. the secondary actors of this systens is payment gateway.

## 4.3 Use Case Diagrams

Use case diagrams give the non-technical view of overall system. In this section use case scenarios are described elaborately.

##### 4.3.1 Level-0 use case:

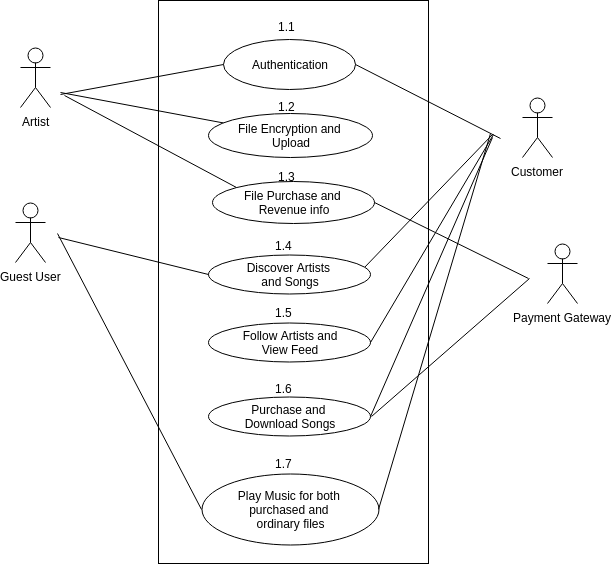


###### Figure 4.1: Level-0 Use case diagram

**System description for level-0 use case diagram:**

|  |  |
| --- | --- |
| **Use case ID: 0** | |
| Primary actor | Artist, Customer, Guest User |
| Goal in context | Encryption based Audio File Distribution System with Compatible Music Player |
| Scenario | There are four actors in system. Three of them are primary. |
| Secondary actor | Payment Gateway |

##### 4.3.2 Level-1 use case:



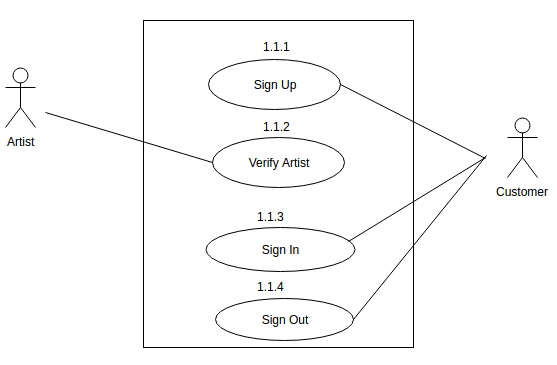
###### Figure 4.2: Level-1 Use case diagram

###### 

###### System description for level-1 use case diagram:

|  |  |
| --- | --- |
| **Use case ID: 1** | |
| Primary actor | Artist, Customer, Guest User |
| Goal in context | Encryption based Audio File Distribution System with Compatible Music Player |
| Scenario | There are four actors in system. Three of them are primary. |
| Secondary actor | Payment Gateway |

##### 4.3.3.1 Subsystems of Level-1.1 Use Case



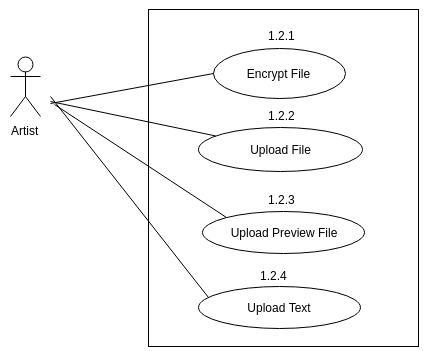
###### Figure 4.3: Level-1.1 Use Case Diagram

###### System description for level-1.1 use case diagram:

|  |  |
| --- | --- |
| **Use case ID: 1.1** | |
| Primary actor | Artist, Customer |
| Goal in context | Authentication |
| Scenario | Artist and Customer sign up, sign in and sign out |
| Secondary actor | None |

##### 

##### 4.3.3.2 Subsystems of Level-1.2 Use Case

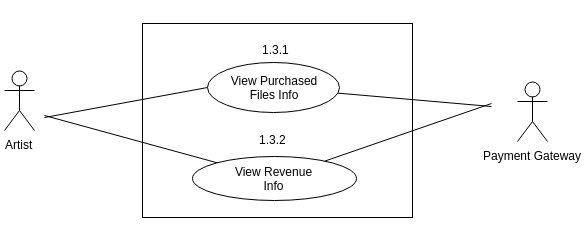


###### Figure 4.4: Subsystems of Level-1.2 Use Case Diagram

###### System description for level-1.2 use case diagram:

|  |  |
| --- | --- |
| **Use case ID: 1.2** | |
| Primary actor | Artist |
| Goal in context | File Encryption and Upload |
| Scenario | Artist encrypt file, upload both the file and preview file with a text |

##### 4.3.3.3 Subsystems of Level-1.3 Use Case



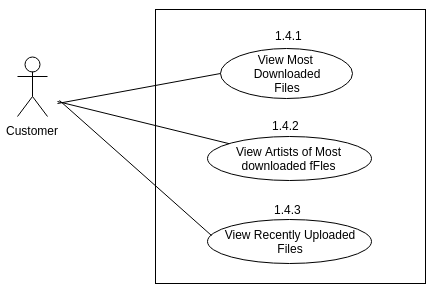
###### Figure 4.5: Subsystems of Level-1.3 Use Case Diagram

###### System description for level-1.3 use case diagram:

|  |  |
| --- | --- |
| **Use case ID: 1.3** | |
| Primary actor | Artist |
| Goal in context | view File purchase and revenue info |
| Scenario | Artist view the numbers of his purchased files by customers and his revenues foreach​ ​songs.  A server database will provide data for payment gateway about every artist’s revenue |
| Secondary actor | Payment Gateway |

##### 

##### 4.3.3.4 Subsystems of Level-1.4 Use Case

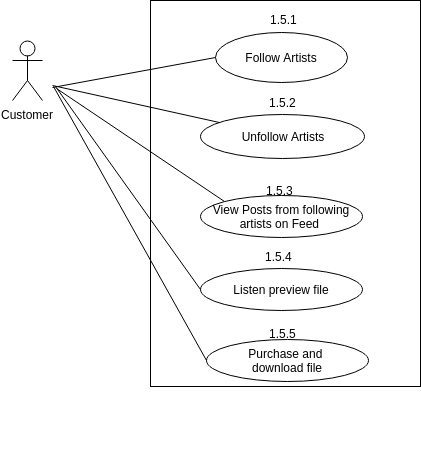


###### Figure 4.6: Subsystems of Level-1.4 Use Case Diagram

###### System description for level-1.4 use case diagram:

|  |  |
| --- | --- |
| **Use case ID: 1.4** | |
| Primary actor | Customer |
| Goal in context | DIscover Artists and songs |
| Scenario | Customer will view top artist list according to artists’ downloaded song numbers, top downloaded file list and most recent uploaded songs list and​ ​ he/she​ ​ can​ ​ download​ ​ and​ ​ purchase​ ​ song​ ​ from​ ​ this​ ​ lists. |
| Secondary actor | None |

##### 4.3.3.5 Subsystems of Level-1.5 Use Case

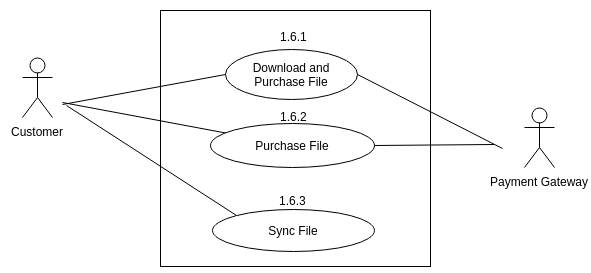


###### Figure 4.7: Subsystems of Level-1.5 Use Case Diagram

###### System description for level-1.5 use case diagram:

|  |  |
| --- | --- |
| **Use case ID: 1.5** | |
| Primary actor | Customer |
| Goal in context | Follow artists and view feed |
| Scenario | There will be profile page for every artists. Customers have option to follow and unfollow  artists​ ​ and​ ​ download​ ​ the​ ​ published​ ​ songs​ ​ from​ ​ every​ ​ artist’s​ ​ profile.  A Customer will view the posts (based on time) from his followed artists on feed​ ​ tab.​ ​ |
| Secondary actor | None |

##### 4.3.3.6 Subsystems of Level-1.6 Use Case



###### 

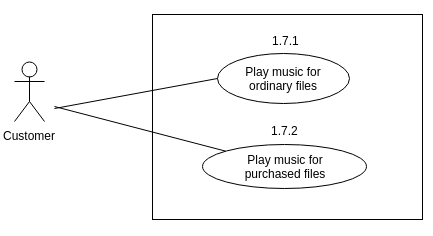
###### Figure 4.8: Subsystems of Level-1.6 Use Case Diagram

###### System description for level-1.6 use case diagram:

|  |  |
| --- | --- |
| **Use case ID: 1.6** | |
| Primary actor | Customer |
| Goal in context | Purchase and download songs |
| Scenario | There will be profile page for every artists. Customers have option ​ ​ download​ ​ the​ ​ published​ ​ songs​ ​ from​ ​ every​ ​ artist’s​ ​ profile.  He/she​ ​ can​ ​ purchase​ ​ any​ ​ files​ ​ produced​ ​ by​ ​ any​ ​ artists​ ​ for​ ​ upto​ ​ 3 ​ ​ devices.  A server database will provide data for payment gateway about every​ ​ customer’s​ ​ usage. |
| Secondary actor | Payment Gateway |

##### 

##### 4.3.3.7 Subsystems of Level-1.7 Use Case



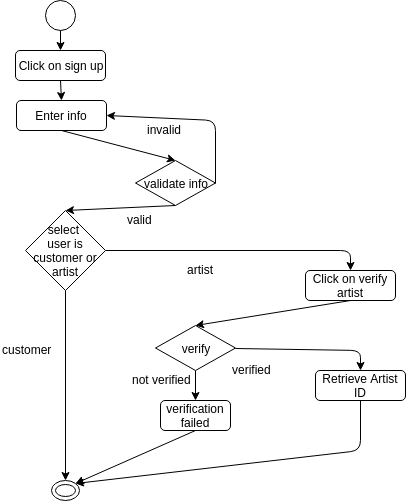
###### Figure 4.9: Subsystems of Level-1.7 Use Case Diagram

###### System description for level-1.7 use case diagram:

|  |  |
| --- | --- |
| **Use case ID: 1.7** | |
| Primary actor | Customer |
| Goal in context | Play music files for both purchased and ordinary files |
| Scenario | A music player tab, which will deliver the functionality of basic music player  for​ ​ the​ ​ both​ ​ the​ ​ purchased​ ​ files​ ​ and​ ​ ordinary​ ​ .mp3​ ​ files. |
| Secondary actor | None |

## 4.4 Activity & Swimlane Diagrams

Activity diagram shows the technical view of the system for every use case from which we can understand how the system actually works and how the actors interact with the system. Swimlane diagram of a specific activity diagram shows the responsibilities of each actor dividing them into lanes.

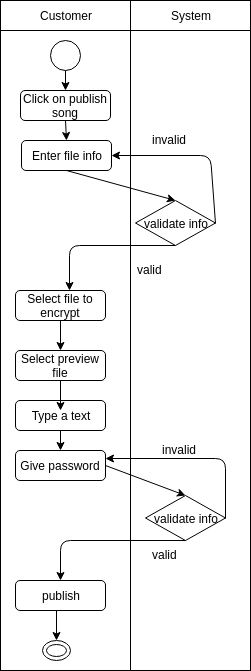
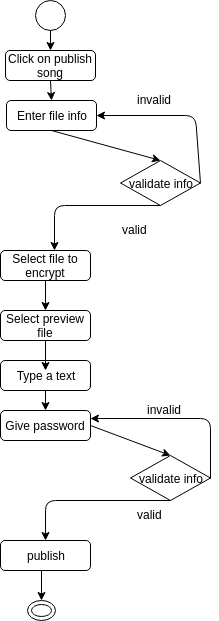


###### Figure 4.10: Activity Diagram for use case 1.1.1

###### swimlane1.1.1.png

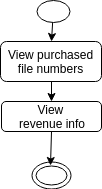
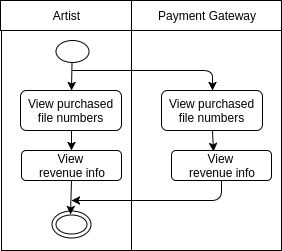
###### 

###### Figure 4.11: Swimlane Diagram for use case 1.1.1



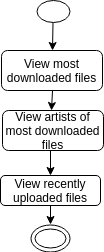
###### Figure 4.12: Activity Diagram Figure 4.13: Swimlane Diagram

###### for use case 1.1.2 for use case 1.1.2

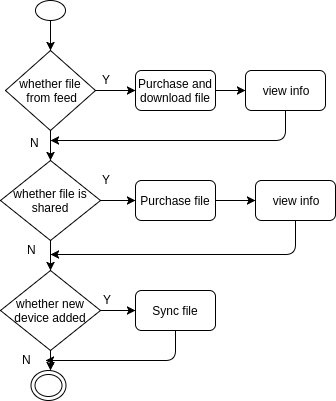
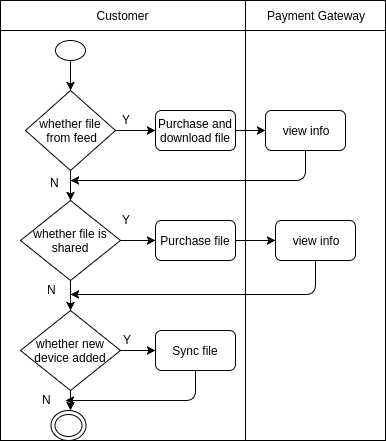
###### Figure 4.14: Activity Diagram Figure 4.15: Swimlane Diagram

###### for use case 1.1.3 for use case 1.1.3

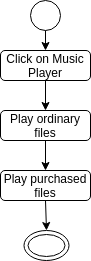
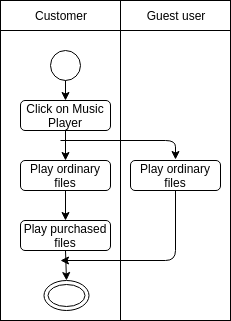
 

###### Figure 4.16: Activity Diagram Figure 4.17: Activity Diagram

###### for use case 1.1.4 for use case 1.1.5

###### Figure 4.18: Activity Diagram and Swimlane Diagram for use case 1.1.6

###### Figure 4.19: Activity Diagram and Swimlane Diagram for use case 1.1.7

### Chapter 5: Data Model

In this chapter I will discuss about the data models of our system.

## 5.1 Data Modeling Concept

If software requirements include the need to create, extend, or interface with a database or if complex data structures must be constructed and manipulated, a software team may choose to create a data model as part of overall requirements modeling.

## 5.2 Data Objects

A data object is representation of composite information that must be understood by software. Here, composite information means that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

##### 5.2.1 Identify Data Objects

Nouns having attributes are selected as data object. Those who doesn’t have any attributes have covered under the data objects.

**Data Object:** Customer

**Attributes:**

* Customer id
* Password
* Name
* Phone Number
* Device id
* Following Artists

**Data Object:** Artist

**Attributes:**

* Artist id
* Password
* Name
* Number of Uploaded Files

**Data Object:** Revenue

**Attributes:**

* Artist id
* Number of artist’s purchased file

**Data Object:** Usage

**Attributes:**

* Customer id
* File id
* Number of purchased files by customer

**Data Object:** File

**Attributes:**

* File id
* Artist id
* Customer id
* Content URL

**Data Object:** Post

**Attributes:**

* Post id
* Artist id
* File id
* Preview file id
* Text

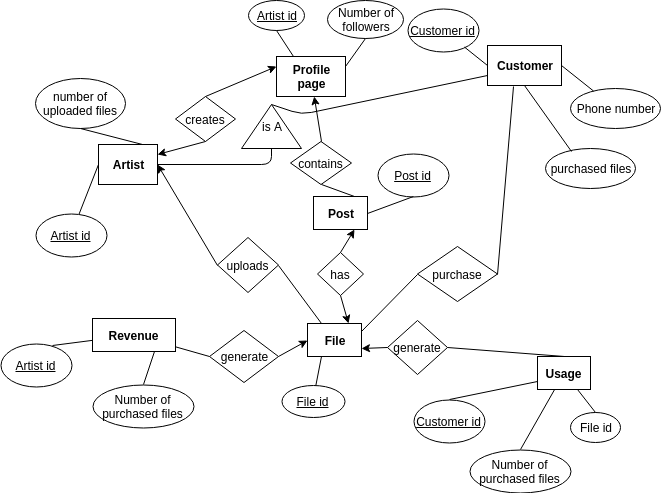
**Data Object:** Profile Page

**Attributes:**

* Number of followers
* Artist id

##### 

##### 5.2.3 Entity Relationship Diagram



###### Figure 5.1: ER Diagram for data based modeling

### Chapter 6: Class Based Model

This Chapter is intended to describe class based modeling of post examination control system.

## 6.1 Class Modeling Concept

Class-based modeling represents the objects that the system will manipulate, the operations that will applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

##### 6.1.1 Identifying Analysis Class

To identify our analysis class we firstly grammatically parsed all the nouns and then categorized them according to general classification and selection criteria.

Following are the steps we used to analysis the classes for our system.

Preliminary Classes:

1. File
2. Post
3. Artist Profile
4. Customer
5. Artist
6. Database
7. Music Player

Preliminary classes & their attributes:

|  |  |
| --- | --- |
| **Classname** | **Attributes** |
| File | file\_id, artist\_id, customer\_id, content |
| Post | artist\_id, File, date\_time, text, raw\_file |
| Artist Profile | artist\_id, follower\_no, Post(s) |
| Customer | customer\_id, password, purchased\_file\_id(s)  phone\_no, usage |
| Artist | artist\_id, uploaded file\_no, revenue |
| Database | All |
| Music Player | customer\_id |

Preliminary classes & their methods:

|  |  |
| --- | --- |
| **Classname** | **Methods** |
| File | encryptFile()  purchaseAndDownloadFile()  purchaseFile()  syncFile() |
| Post | playPreviewFile()  showOnFeed() |
| Artist Profile | showPosts()  showFollowerNo()  follow()  unfollow() |
| Customer | signUp()  signIn()  signOut()  viewUsageInfo() |
| Artist | requestVerifyForArtist()  uploadPost()  viewRevenueInfo() |
| Database | create()  update()  viewCustomerUsageInfo()  viewArtistRevenueInfo() |
| Music Player | playOrdinaryFiles()  playPurchasedFiles() |

## Class Cards:

|  |  |
| --- | --- |
| **File** | |
| **Attributes** | **Methods** |
| file\_id  artist\_id  customer\_id  content | encryptFile()  purchaseAndDownloadFile()  purchaseFile()  syncFile() |
| **Responsibilities** | **Collaborative Class** |
| Encrypting file and other file processings | Artist, Artist profile, Customer |

|  |  |
| --- | --- |
| **Post** | |
| **Attributes** | **Methods** |
| artist\_id  File  date\_time  text  raw\_file | playPreviewFile()  showOnFeed() |
| **Responsibilities** | **Collaborative Class** |
| Proessed in File uploading from artist | File, Artist, Artist Profile |

|  |  |
| --- | --- |
| **Artist Profile** | |
| **Attributes** | **Methods** |
| artist\_id  follower\_no  Post(s) | showPosts()  showFollowerNo()  follow()  unfollow() |
| **Responsibilities** | **Collaborative Class** |
| Provides artist view for customers | Customer, Post, Artist |

|  |  |
| --- | --- |
| **Customer** | |
| **Attributes** | **Methods** |
| customer\_id  password  purchased\_file\_id(s)  phone\_no  usage | signUp()  signIn()  signOut()  viewUsageInfo() |
| **Responsibilities** | **Collaborative Class** |
| Authenticate, purchases file and view usage | File, Artist Profile |

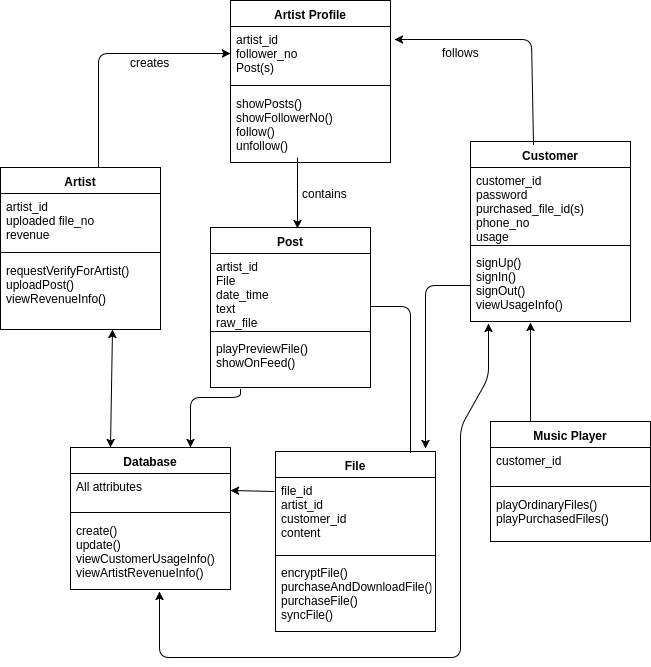
|  |  |
| --- | --- |
| **Artist** | |
| **Attributes** | **Methods** |
| artist\_id  uploaded file\_no  revenue | requestVerifyForArtist()  uploadPost()  viewRevenueInfo() |
| **Responsibilities** | **Collaborative Class** |
| Uploads file and check revenue | Artist Profile, Post |

|  |  |
| --- | --- |
| **Database** | |
| **Attributes** | **Methods** |
| All attributes | create()  update()  viewCustomerUsageInfo()  viewArtistRevenueInfo() |
| **Responsibilities** | **Collaborative Class** |
|  | All except Music Player |

|  |  |
| --- | --- |
| **Music Player** | |
| **Attributes** | **Methods** |
| customer\_id | playOrdinaryFiles()  playPurchasedFiles() |
| **Responsibilities** | **Collaborative Class** |
| Provide music player functionality | Customer, File |

##### 

##### 6.1.2 Class Responsibility Collaboration (CRC) Diagram



###### Figure 6.1: CRC Diagram

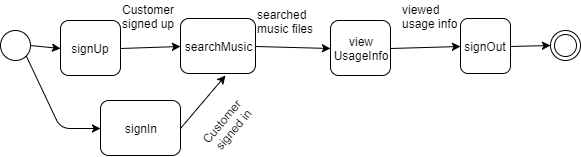
### Chapter 7: Behavioral Model

The behavioral model indicates how software will respond to external events.

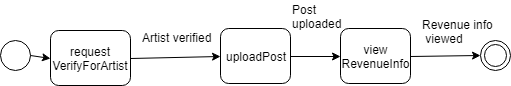
## 7.1 State Transition Diagram

State diagram represents active states for each class the events (triggers). For this we identified all the events, their initiators and collaborators.

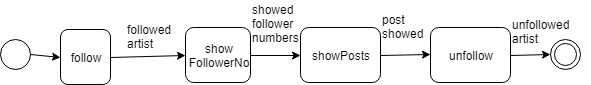
##### State transition diagrams for all classes:



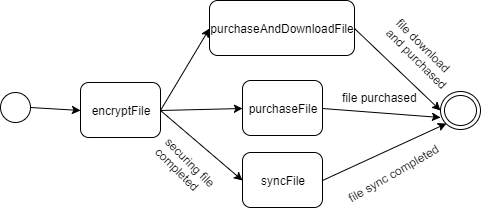
###### Figure 7.1: State Transition Diagram for Customer Class



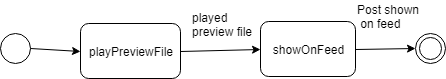
###### Figure 7.2: State Transition Diagram for Artist Class



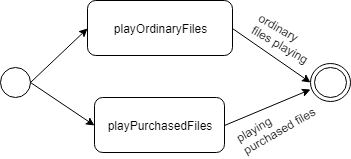
###### Figure 7.3: State Transition Diagram for Artist Profile Class



###### Figure 7.4: State Transition Diagram for File Class



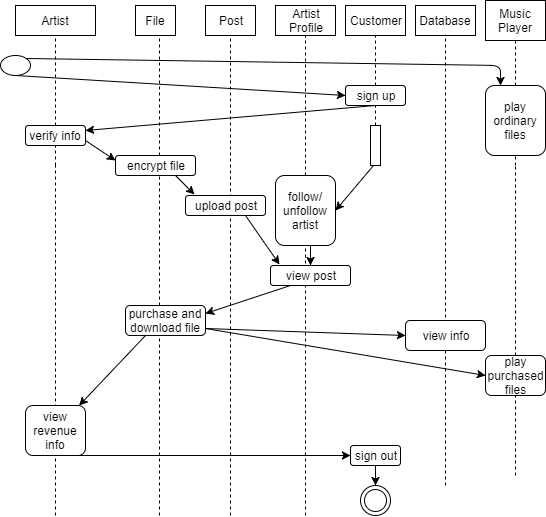
###### Figure 7.5: State Transition Diagram for Post Class



###### Figure 7.6: State Transition Diagram for Music Player Class

## 7.2 Sequence Diagram

Sequence diagram indicates how events cause transitions from object to object.



###### Figure 7.7: Sequence Diagram

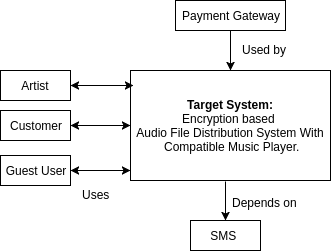
### Chapter 8: Software​ ​Design​ ​Architecture

The behavioral model indicates how software will respond to external events.

## 8.1​ ​Introduction

Software architectural design has been followed in **Encryption based Audio File Distribution System With Compatible Music Player**. At first, the context diagram is defined. Then the archetype of the project is described. And after the finding the archetypes, the components and the classes are also defined.

## 8.2​ ​Representing​ ​the​ ​system​ ​in​ ​context



###### Figure 8.1: Architectural Context Diagram

In the context diagram, the total system has three actors and they are: Artist, Customer and Guest User. Artist and Customer are the primary actor of this system. These actors give input to the system and receive the output from the system.The system uses Internet as subordinate system. Payment Gateway is superordinate system of the target system

## 8.3​ ​Define​ ​archetypes:

1. Authentication
2. File Upload and view revenue info
3. Find artists and posts
4. File purchase and music player

## 8.4​ ​Refining​ ​archetypes​ ​into​ ​components:

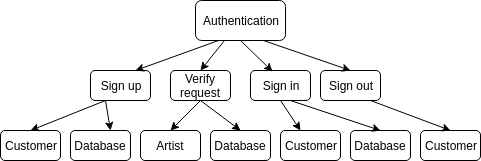
##### Components:

1. Sign up, Sign in, Sign out, verify request, DAL
2. Encrypt file, Upload post, view purchase info, view revenue info, DAL
3. Search, Follow artist, View post, DAL
4. Download and purchase file, sync file, Play ordinary and purchased files, DAL

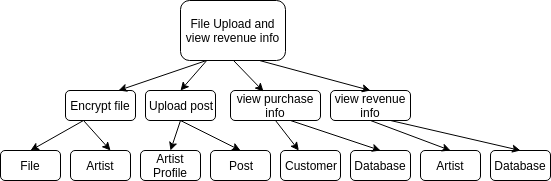
##### Classes:

1. Customer, Artist, Database
2. File, Post, Artist, Customer
3. Artist Profile, Post, File
4. Database, Music Player, File, Customer

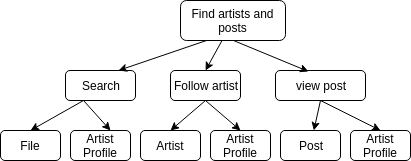
## 8.5 Describing instantiation of the system



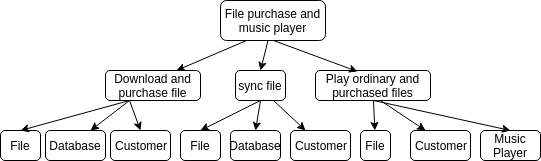
###### Figure 8.2: Refining ‘Authentication’ archetype into components and classes



###### Figure 8.3: Refining ‘File Upload and Revenue info’ archetype into components and classes



###### Figure 8.4: Refining ‘Find artists and posts’ archetype into components and classes



###### Figure 8.5: Refining ‘Find purchase and music player’ archetype into components and classes

### Chapter 9: Component Level Design

This chapter describes the preliminary test plan of Encryption based Audio File Distribution System With Compatible Music Player.

**Step 1:** Identify all design classes that correspond to the problem domain as defined in the analysis model and architectural model.

The classes are:

1. Customer

2. Artist

3. File

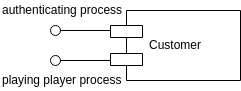
4. Post

**Step 2**: Identify all design classes that correspond to the infrastructure domain. Such classes that are not belong to the problem domain for our project are:

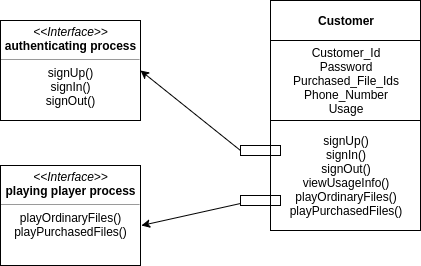
i. DAL (Data Access Layer)

**Step3**: Elaborate all design classes that are not acquired as reusable components

**1.1 Analyzing Class 1.2 Design Component**

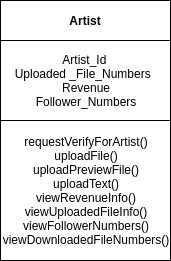
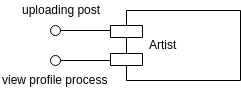
**1.3 Elaborating Class**

****

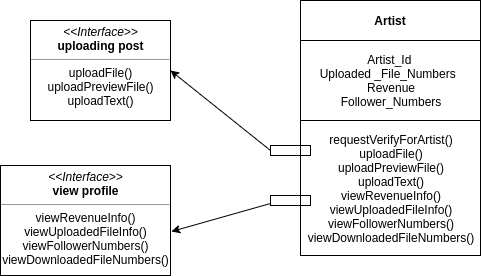
###### 

###### Figure 9.1 Customer Class Component

**1.1 Analyzing Class 1.2 Design Component**

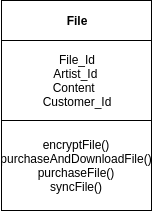
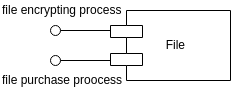
**1.3 Elaborating Class**

****

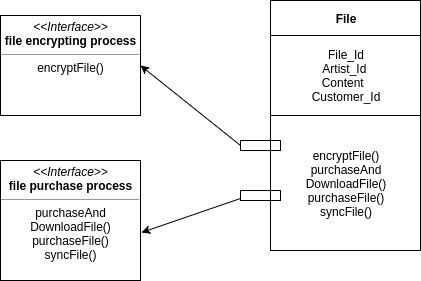
###### 

###### Figure 9.2 Artist Class Component

**1.1 Analyzing Class 1.2 Design Component**

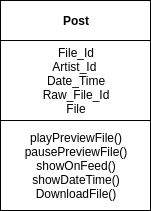
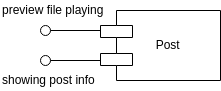
**1.3 Elaborating Class**

****

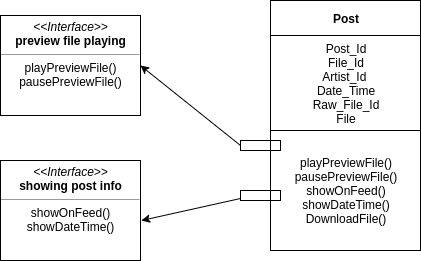
###### 

###### Figure 9.3 File Class Component

**1.1 Analyzing Class 1.2 Design Component**

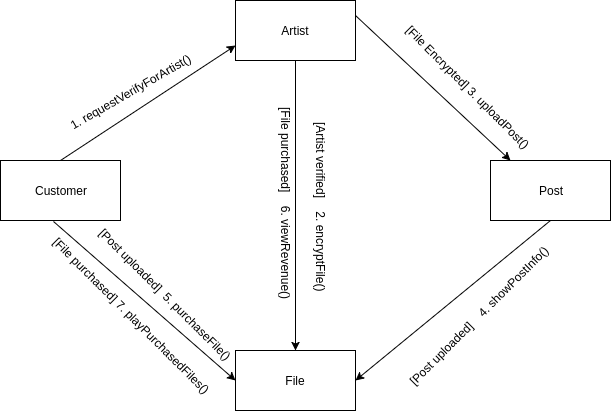
**1.3 Elaborating Class**

****

###### 

###### Figure 9.4 Post Class Component

**Step 3(a): Specify message details when classes or components collaborate**



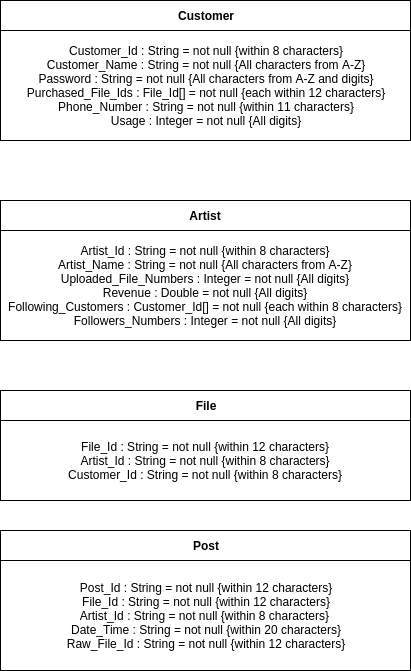
###### Figure 9.5 Specifying message details

**Step 3(b):** I think the elaborated classes need not to be refactored any more. As a result, no

interface is needed for this design.

**Step 3(c):** Elaborate attributes and define data types and data structures required to implement

them.



**Step 3(d):** Describe processing flow within each operation in detail by means of pseudo code or

UML activity diagrams.

**Step 4:**

The persistent data sources (databases and files) and the classes required to manage them are

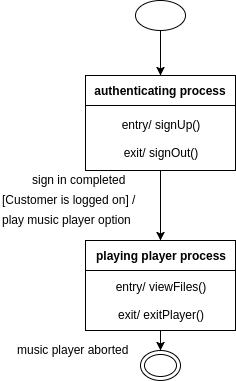
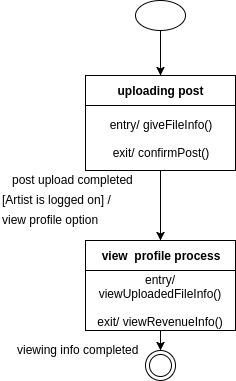
following:

**Persistent data source:** Database

**Classes to manage data source:** Database classes for each entity

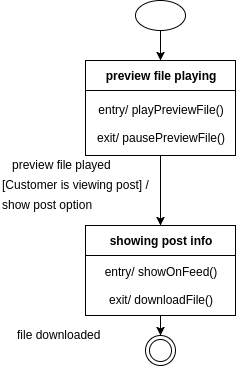
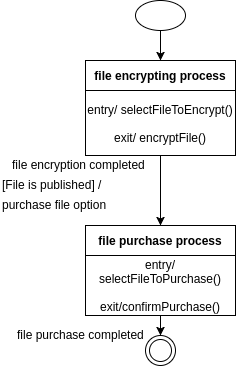
**Step 5:**

Develop and elaborate behavioral representations for a class or component.

###### Figure 9.6 Elaborating behavior Figure 9.7 Elaborating behavior

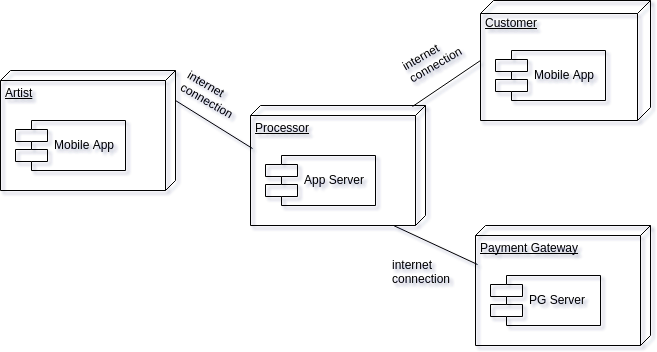
###### for Customer Class for Artist Class

###### Figure 9.8 Elaborating behavior Figure 9.9 Elaborating behavior

###### for Post Class for File Class

**Step 6:** Elaborate deployment diagrams to provide additional implementation detail.



###### Figure 9.10 : Elaborate Deployment

###### 

### Chapter 10: User Interface Design

## 10.1 Introduction

User interface design (UI) or is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. User interface design creates an effective communication medium between a human and a computer. Following a set of interface design principles, design identifies interface objects and actions and then creates a screen layout that forms the basis for a user interface prototype.

## User Analysis

**Target User:** Users of this system are basically the artists (musicians) and who listens music frequently.

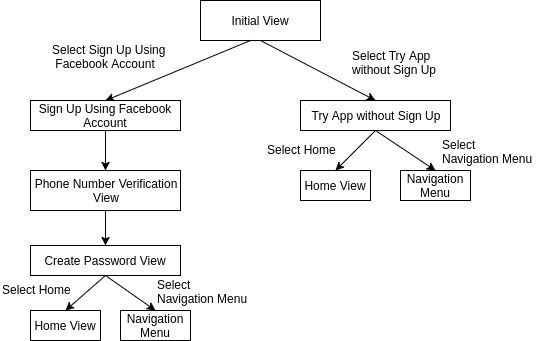
**Knowledge Level:** Average users should have knowledge of English and internet browsing capability.

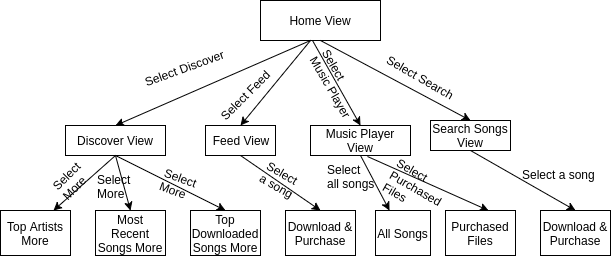
**Age range:** User Community will be between 15-60.

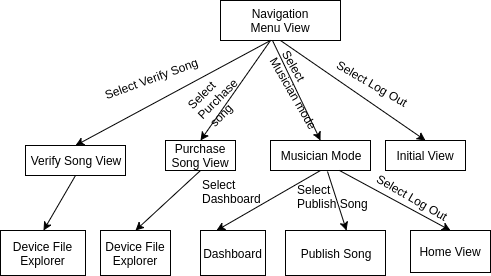
**Gender:** Both male and female.

**Frequency of Use:** All users will use this system frequently.

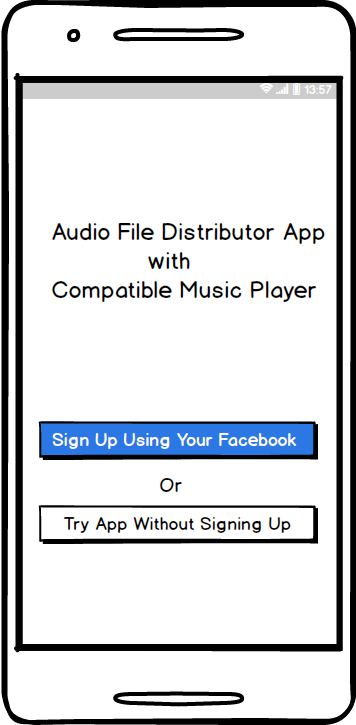
## 10. 2 Event Transition







This is the initial user interface. A user will view this interface whenever he/she will open the application.



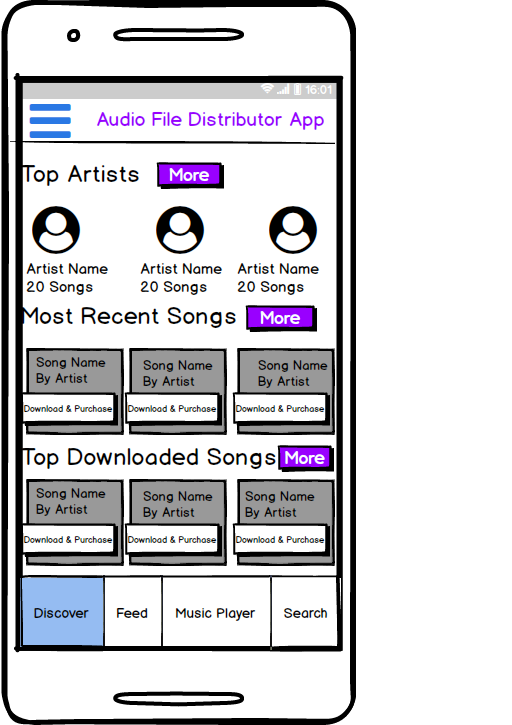
###### Figure 10.1 : User Interface of Initial View

This is the **‘**Phone Number Verification’ interface. A user with verify his/her phone number through this user interface.



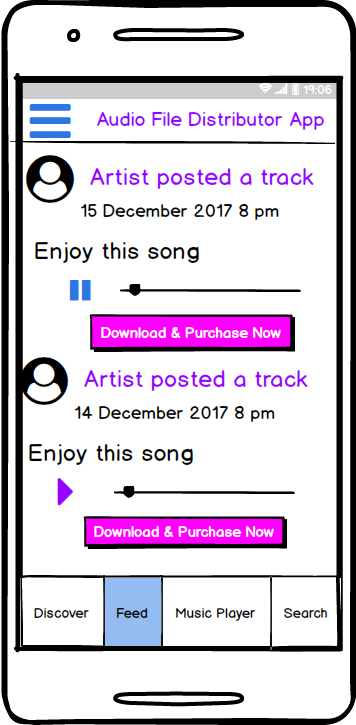
###### Figure 10.2 : User Interface of Phone Number Verification

This “Discover” interface for both customer and guest user. The only difference for these users here is that, guest user user will not be able to download a song file.



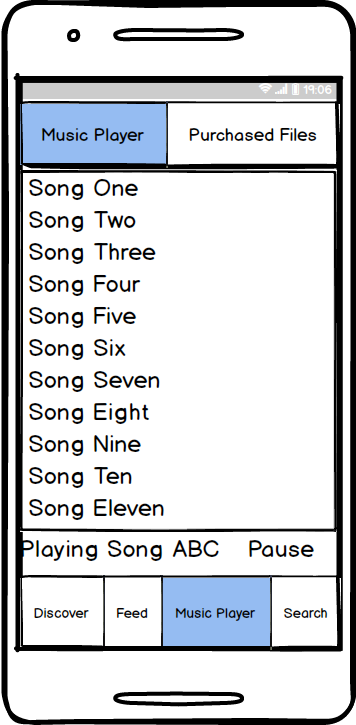
###### Figure 10.3 : User Interface of Discover View

This “Feed” interface for both customer and guest user. The only difference for these users here is that, guest user user will not be able to download a song file.



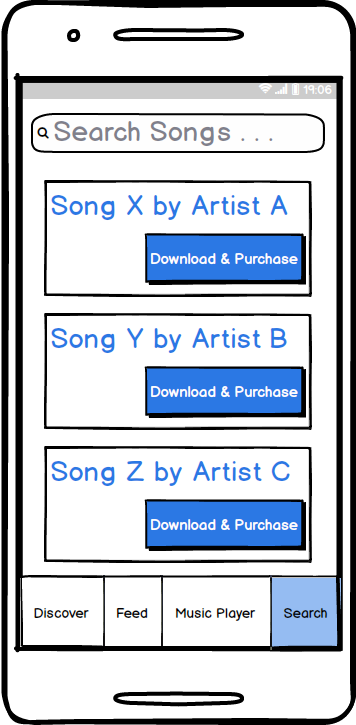
###### Figure 10.4 : User Interface of Feed View

This “Music Player” interface for both customer and guest user. The only difference for these users here is that, guest user user can play only ordinary .mp3 files.



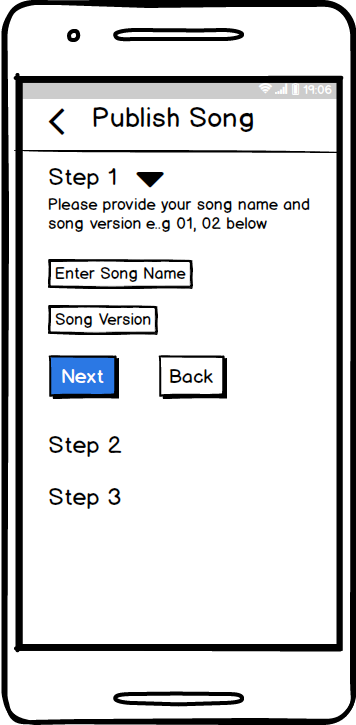
###### Figure 10.5 : User Interface of Music Player View

This “Search” interface for both customer and guest user. The only difference for these users here is that, guest user user will not be able to download a song file.



###### Figure 10.6 : User Interface of Search View

This “Publish Song Step 1” interface is for artist. He/she will first Enter song name and song version name to generate a file id for for this file.



###### Figure 10.7 : User Interface of Publish Song Step 1 View

This “Publish Song Step 2” interface is for artist. He/she will select a file from his device to encrypt and publish.



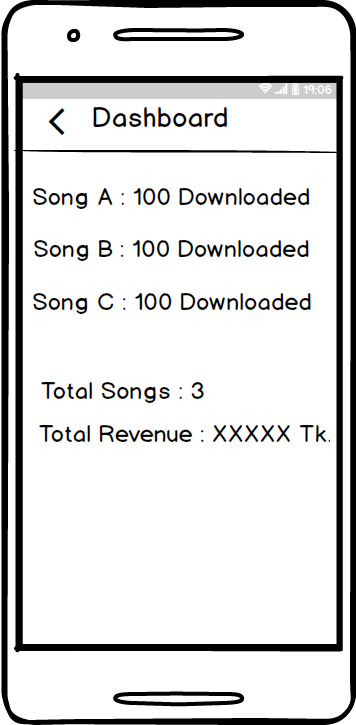
###### Figure 10.8 : User Interface of Publish Song Step 2 View

This “Publish Song Step 3” interface is for artist. He/she will write some text and select a file as preview file from his device to publish.



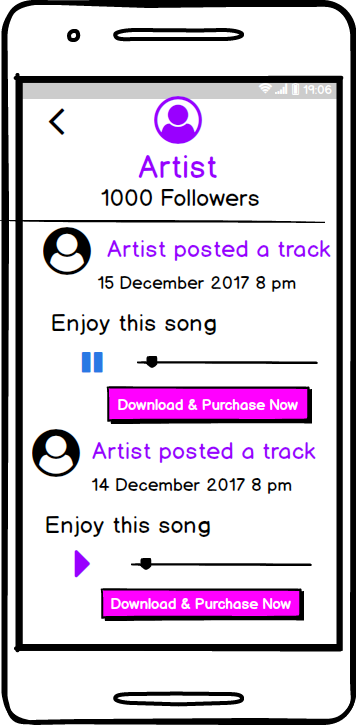
###### Figure 10.9 : User Interface of Publish Song Step 3 View

This “Dashboard” interface is for artist. He/she will view information about his/previously uploaded files here.



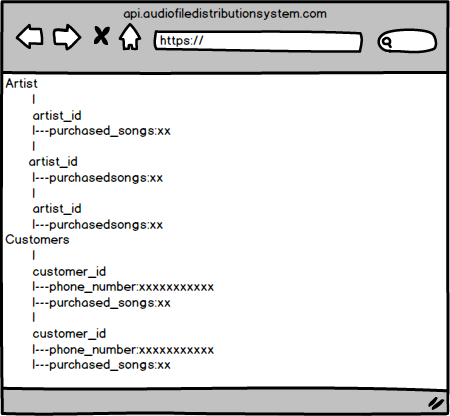
###### Figure 10.10 : User Interface of Dashboard View

This “Artist Profile” interface is for customers and guest users. The only difference for these users here is that, guest user user will not be able to download a song file.



###### Figure 10.11 : User Interface of Artist Profile View

This “Payment Gateway” interface is for Payment Gateway. It is a web interface for consuming necessary information of artists and customers.



###### Figure 10.12 : Web User Interface of Payment Gateway

### Chapter 11: Test Plan and Test Cases

This chapter describes test plan and test cases of Encryption based Audio File Distribution System With Compatible Music Player (EAFDSCMP).

## 11.1 Test Plan Identifier

The Test Plan has been created to communicate the test approach to readers of the document. This is the Master Test Plan of my project. As the software will be released once I will stick to the master test plan. The id of this test plan is EAFDSCMP-0616-v-1.0

## 11.2 Introduction

This test plan describes the testing approach and overall framework that will drive the testing of the EAFDSCMP-0616-v-1.0. The Test Plan document documents and tracks the necessary information required to effectively define the approach to be used in the testing of the project’s product. The Test Plan document is created during the Planning Phase of the project. Its intended audience is the project supervisor, developer and testing team. Some portions of this document may on occasion be shared with the client/user and other stakeholder whose input/approval into the testing process is needed.

##### 11.2.1 Summary of items and features to be tested

Mobile application version is to be tested for EAFDSCMP-0616-v-1.0. In this application, it has some features that to be tested, these are: Authentication, File Encryption and Upload, File Purchase and Revenue info, Discover Artists and Songs, Follow Artists and View Feed, Purchase and Download Songs, Play Music for both purchased and ordinary files.

##### 11.2.2 Requirement and history of items

The requirement of items that will be tested collect from the software requirement and specification (SRS) and architectural design primarily.

##### 11.2.3 High level description of testing goals

Testing makes a software bug free and more secure. After releasing a software product, there may have a lot of bugs which may not be solved during the development phase. These bugs are solved after testing phase. So after development, testing is required for developing a better software product.

##### 11.2.4 Reference Document

* Software requirement and specification
* Architecture design
* Component design
* User Interface design

## 11.3 Test items

The following parts of EAFDSCMP-0616-v-1.0 will be tested:

1. Encryption Based Audio File Distribution System With Compatible Music Player

## 11.4 Features to be tested

1. Sign up
2. Sign In
3. Sign out
4. Verify request
5. Encrypt file and Upload Post
6. View purchase and revenue info
7. Search Songs and artists
8. Follow & Unfollow Artist
9. View post
10. Sync file
11. Play ordinary and purchased files

## 11.5 Features not to be tested

All the defined features will be tested. Therefore, no feature will exist that will not be tested.

## 11.6 Approach

* I will cover only black box testing for my project.
* I will test the functionality of the application.
* I will test all the features in Xiaomi Redmi Note 3 (Android Marshmallow edition) smartphone only.
* Integration testing will be performed.

## 11.7 Item pass/fail criteria

Specifying the criteria that I will use to determine whether each test item of my project has passed or failed during testing. The planning criteria gives the framework for how the system will be evaluated and under what circumstances it will be released.

## 11.8 Test Deliverables

I will provide the following deliverables after testing at the end of the project:

* Test plan document
* Test cases

## 11.9 Test Cases

**Test Case 1:**

**Test case Name:** Sign Up

**Short Description:** Test “Sign Up Using Facebook”, “Phone Number Verification”,  “Create Password” and “Try App Without Signing up”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test steps** | **Action** | **Pre-condition** | **Input test data** | **Expected system response** | **Pass/**  **Fail** |
| 1. | “Sign Up Using Facebook” button click event with input data | User is not signed up | Facebook username and password | Enter into logged in state and show “phone number verification” view | Pass |
| 2. | “Phone Number Verification” button click event  with valid phone number  and correct short code | “Sign Up Using Facebook” step is completed | A valid phone number  and correct short code | After confirming phone no, user receives a verification code and entering the code gives message “Verified” | Pass |
| 3. | “Phone Number Verification” button click event  with invalid phone number | “Sign Up Using Facebook” step is completed | An invalid phone number | Does not allow to confirm phone number | Pass |
| 4. | “Phone Number Verification” button click event  with valid phone number and incorrect shortcode | “Sign Up Using Facebook” step is completed | An valid phone number and incorrect shortcode | Show message “Wrong code. Please Re-enter” | Pass |
| 5. | “Create Password” confirm button click event  with two matched password | “Phone Number Verification”  step is completed | 1st input = 1234  2nd input  = 1234 | Shows message “Password created ” | Pass |
| 6. | “Create Password” confirm button click event  with two unmatched password | “Phone Number Verification”  step is completed | 1st input = 1234  2nd input  = 5679 | Shows message  “Password NOT matched. Try again” | Pass |
| 7. | “Create Password” confirm button click event  with two password length less than 4 | “Phone Number Verification”  step is completed | 1st input = 123  2nd input  = 123 | Does not allow to confirm password | Pass |
| 8. | “Try app without sign up” button click event | User is not signed up | Not applicable | Enter into  Guest User state  and show “Discover” view | Pass |

**Test Case 2:**

**Test case Name:** Sign In

**Short Description:** Test Sign In

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test steps** | **Action** | **Pre-condition** | **Input test data** | **Expected system response** | **Pass/**  **Fail** |
| 1. | App restart event after Signing Up | User is signed up  previously | Not applicable | Enter into logged in state as customer and show “Discover” view | Pass |

**Test Case  3:**

**Test case Name:** Sign Out

**Short Description:** Test Sign Out

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test steps** | **Action** | **Pre-condition** | **Input test data** | **Expected system response** | **Pass/**  **Fail** |
| 1. | “Logout” button clicking as guest user | User is logged in as guest user | Not applicable | Go back to Initial Sign Up activity | Pass |
| 2. | “Logout” button clicking as customer | User is logged in as customer | Not applicable | Go back to Initial Sign Up activity | Pass |

**Test Case 4:**

**Test case Name:** Verify Request

**Short Description:** Test Verify Request

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test steps** | **Action** | **Pre-condition** | **Input test data** | **Expected system response** | **Pass/**  **Fail** |
| 1. | “Musician Mode” button clicking as guest user | User is logged in as guest user | Not applicable | Show message  “Sorry! Please sign up first” | Pass |
| 2. | Clicking “Musician Mode” and confirm “Verify Request” | Customer is not verified as artist | Not applicable | Show dialog to verify as artist  and  clicking “OK”, show “You Verified” message.  Clicking “Cancel” discards the dialog. | Pass |
| 3. | “Musician Mode” button clicking as customer  verified previously  as artist | User is logged in as customer and  verified previously  as artist | Not applicable | Show a new activity of artist | Pass |

**Test Case 5:**

**Test case Name:** Encrypt File and Upload Post

**Short Description:** Test Encrypt File and Upload Post

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test steps** | **Action** | **Pre-condition** | **Input test data** | **Expected system response** | **Pass/**  **Fail** |
| 1. | Song Title  character length < 10  And version character length < 2 | Must be logged in as artist | Song Title  text length < 10  And version text length < 2 | Show message  “Song Title  text length must be 20 character long  And version text length must be 2 | Pass |
| 2. | Song Title  character length >= 10  And version character length = 2 | Must be logged in as artist | Song Title  text length >= 10  And version text length = 2 | Go to Step 2 | Pass |
| 3. | Select File  And Encryption successful | Must be logged in as artist | An Audio File from device file explorer | Show message  “Encryption successful”  And go to Step 3 | Pass |
| 4. | Select File  And Encryption failed | Must be logged in as artist | A large File (>80 MB) from device file explorer | Show message  “Encryption failed. Try again” | Pass |
| 5. | “Upload Preview file and complete process” click event  And Upload Successful | Must be logged in as artist and complete Step 1 and Step 2 | An Audio File from device file explorer | Show message  “Upload successful” | Pass |
| 6. | “Upload Preview file and complete process” click event  And Upload Failed | Must be logged in as artist and complete Step 1 and Step 2 | An Audio File from device file explorer | Show message  “Upload failed. Try again” | Pass |

**Test Case 6:**

**Test case Name:** View Purchase Info

**Short Description:** Test Purchase Info

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test steps** | **Action** | **Pre-condition** | **Input test data** | **Expected system response** | **Pass/**  **Fail** |
| 1. | “Download and Purchase” button clicking as guest user | Button is clicked from either in feed tab or search bar tab | Not applicable | Show message  “Sorry! Please sign up first” | Pass |
| 2. | “Download and Purchase” button clicking as customer | Button is clicked from either in feed tab or search bar tab | Customer Password | Show a confirmation dialog to continue and after confirming start downloading, then show message “purchase successful” | Pass |
| 5. | “Purchase New Song” button clicking as guest user | User is logged in as guest user | An Audio File from device file explorer | Show message  “Sorry! Please sign up first” | Pass |
| 6. | “Purchase New Song” button clicking as customer | App must be opened and user is logged in as customer | An Audio File from device file explorer | Show a file explorer dialog to choose a file from storage of device and after choosing a file show purchase complete message | Pass |

**Test Case 7:**

**Test case Name:** Search Songs and artists

**Short Description:** Test Search Songs and artists

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test steps** | **Action** | **Pre-condition** | **Input test data** | **Expected system response** | **Pass/**  **Fail** |
| 1. | Search button click event | App must be opened | Not applicable | Show search bar | Pass |
| 2. | Search bar click event  with empty name | Search button must be clicked | “”  or “ “ or “    “ | Show empty list | Pass |
| 3. | “Search bar click event | Search button must be clicked | “DI” or  “di” or”Di” or “di” | Show list of songs which name starts with input text | Pass |

**Test Case  8:**

**Test case Name:** Follow and Unfollow Artist

**Short Description:** Test Follow and Unfollow Artist

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test steps** | **Action** | **Pre-condition** | **Input test data** | **Expected system response** | **Pass/**  **Fail** |
| 1. | “Follow ” click event in an artist’ profile | Must be logged in as customer | Not applicable | Changed button  “Following” | Pass |
| 2. | “Unfollow ” click event in an artist’ profile | Must be logged in as customer | Not applicable | Changed button  “Follow” | Pass |

**Test Case  9:**

**Test case Name:** View Post

**Short Description:** Test View Post

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test steps** | **Action** | **Pre-condition** | **Input test data** | **Expected system response** | **Pass/**  **Fail** |
| 1. | Feed button click event by guest user | User is logged in as guest user | Not applicable | Show list of predefined posts by artists | Pass |
| 2. | Feed button click event by customer  Without following any artist | Feed button click event by customer | Not applicable | Feed button click event by customer  Without following any artist | Pass |
| 3. | Feed button click event by customer | User is logged in as customer and following at least one artist | Not applicable | Show list of  posts by following artists | Pass |

**Test Case  10:**

**Test case Name:** Sync File

**Short Description:** Test Sync File

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test steps** | **Action** | **Pre-condition** | **Input test data** | **Expected system response** | **Pass/**  **Fail** |
| 1. | “Sync This Device” button clicking as customer | User is logged in as customer | Customer password | Clicking “Verify Song” shows that every purchased file contains device’s IMEI number. | Pass |

**Test Case  11:**

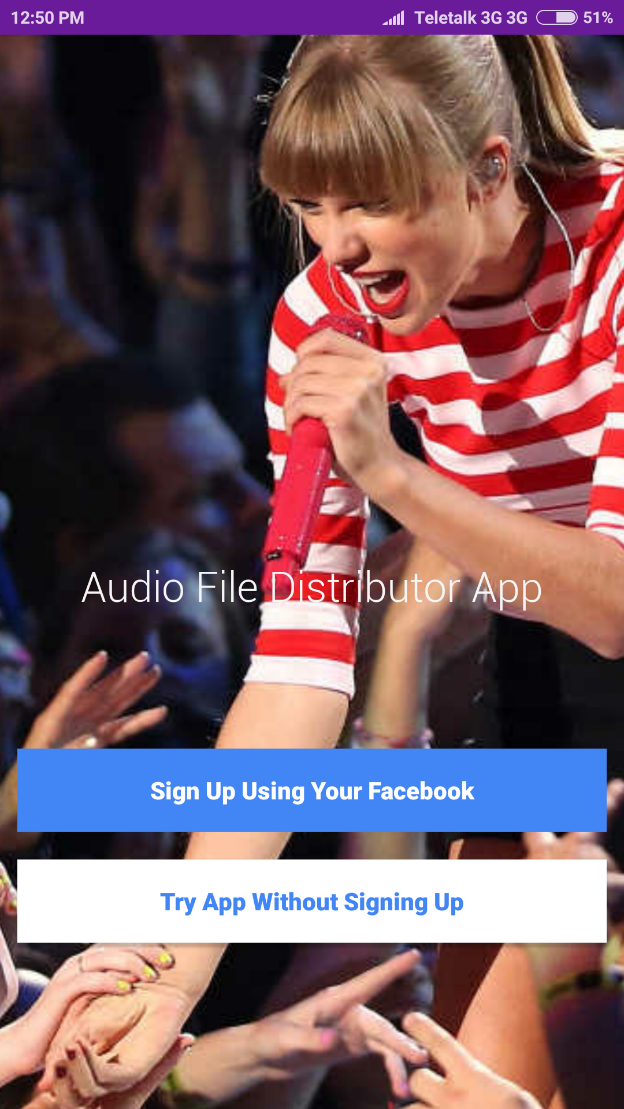
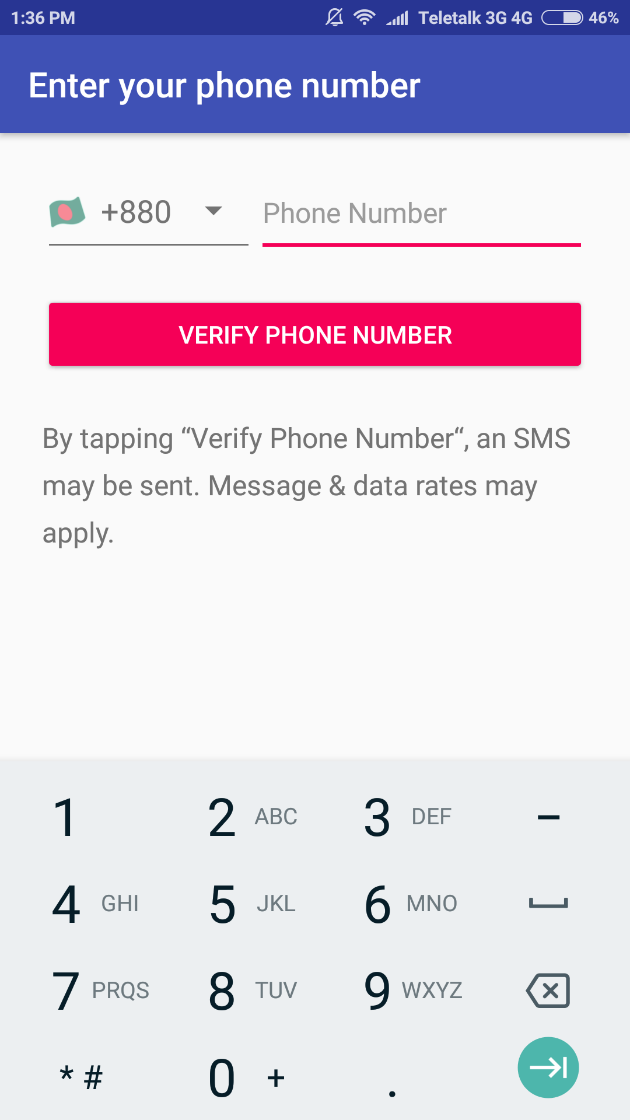
**Test case Name:** Play ordinary and purchased files

**Short Description:** Test Playing ordinary and purchased files

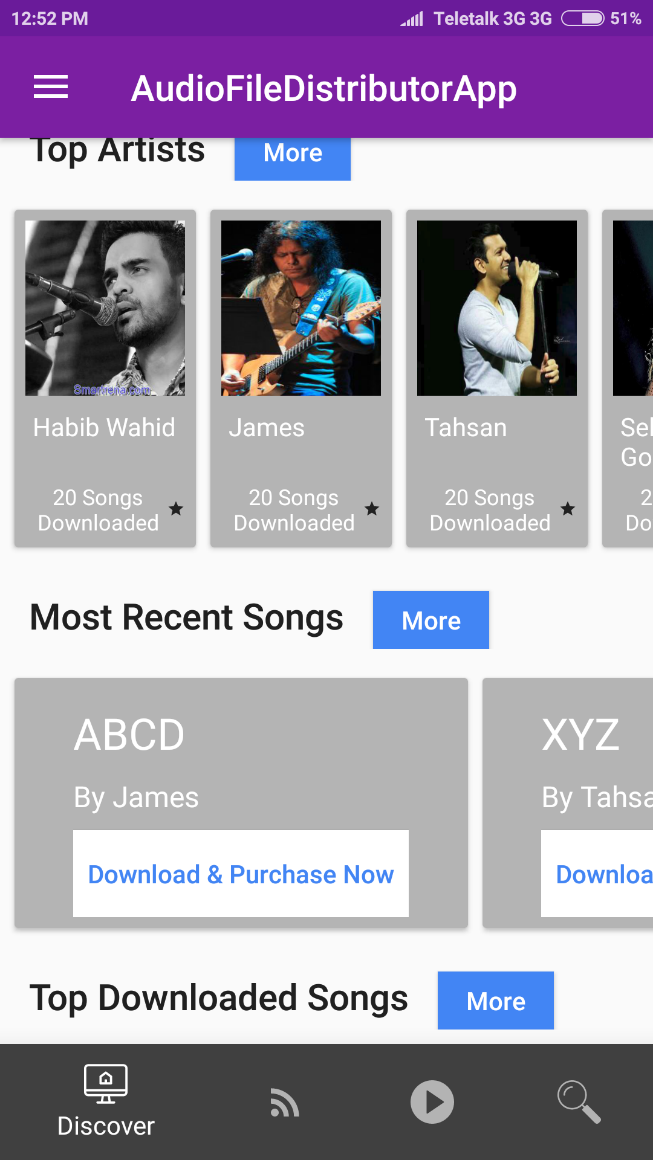
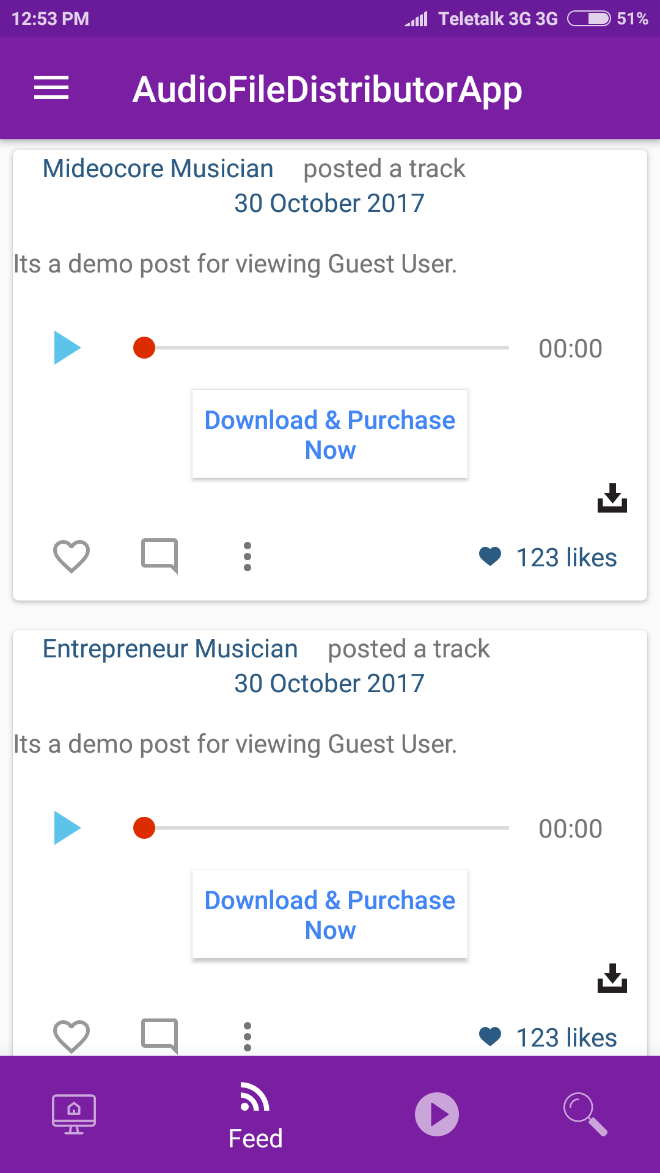
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test steps** | **Action** | **Pre-condition** | **Input test data** | **Expected system response** | **Pass/**  **Fail** |
| 1. | Music Player button click event | User is logged in as guest user | Not applicable | Show Music Player having ordinary file lists and purchased file lists on different tabs | Pass |
| 2. | Ordinary file playing event | Music Player View is opened | An mp3 audio file from list | File play and pause event works | Pass |
| 3. | Purchased file playing event | Music Player View is opened | Purchased encrypted file | File play and pause event works | Pass |

### Chapter 12: User Manual

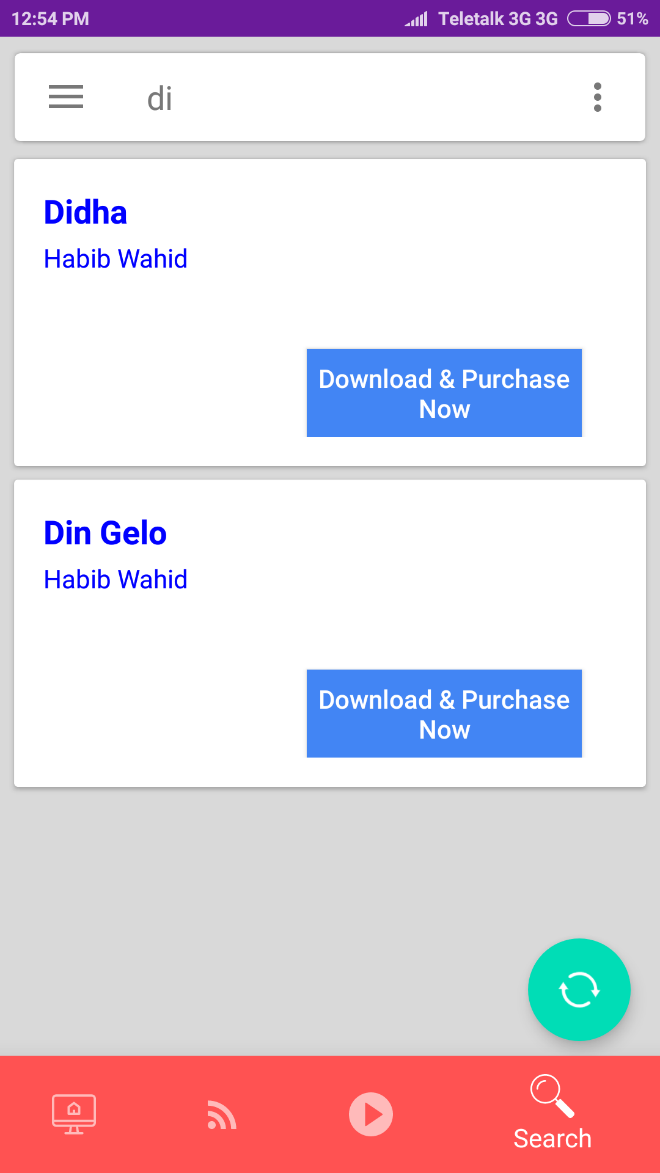
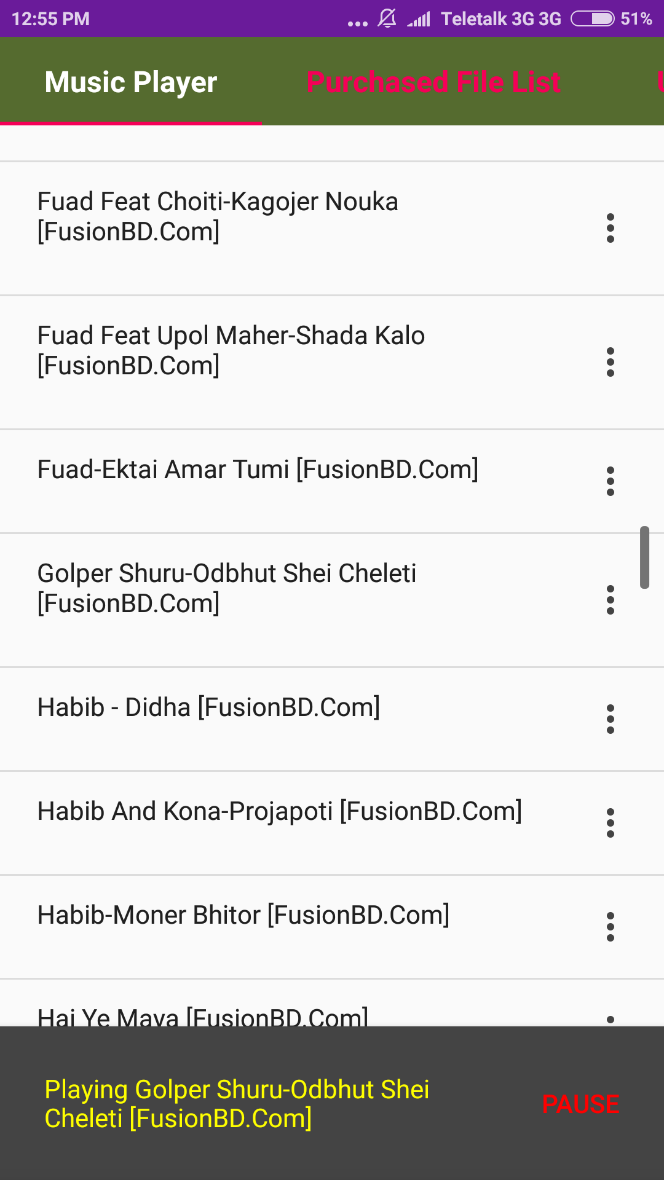
User manual consists of the implemented user interface with information of how a user can interact with it. The first screenshot shows the Initial page. After Sign up completion, a user will verify his/her phone number (2nd screenshots).

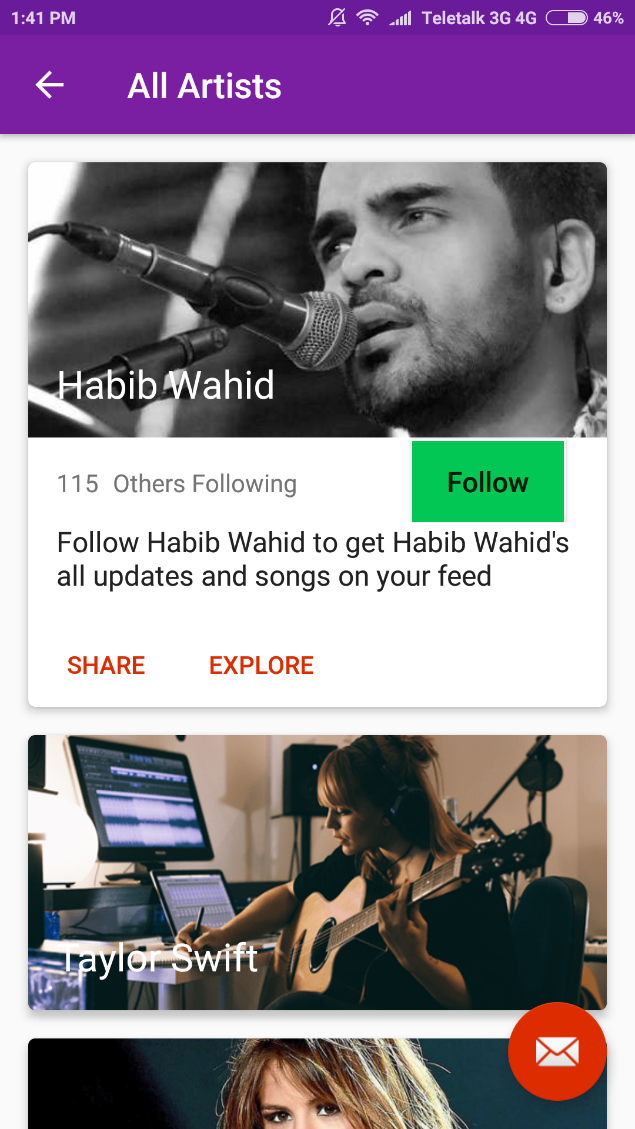
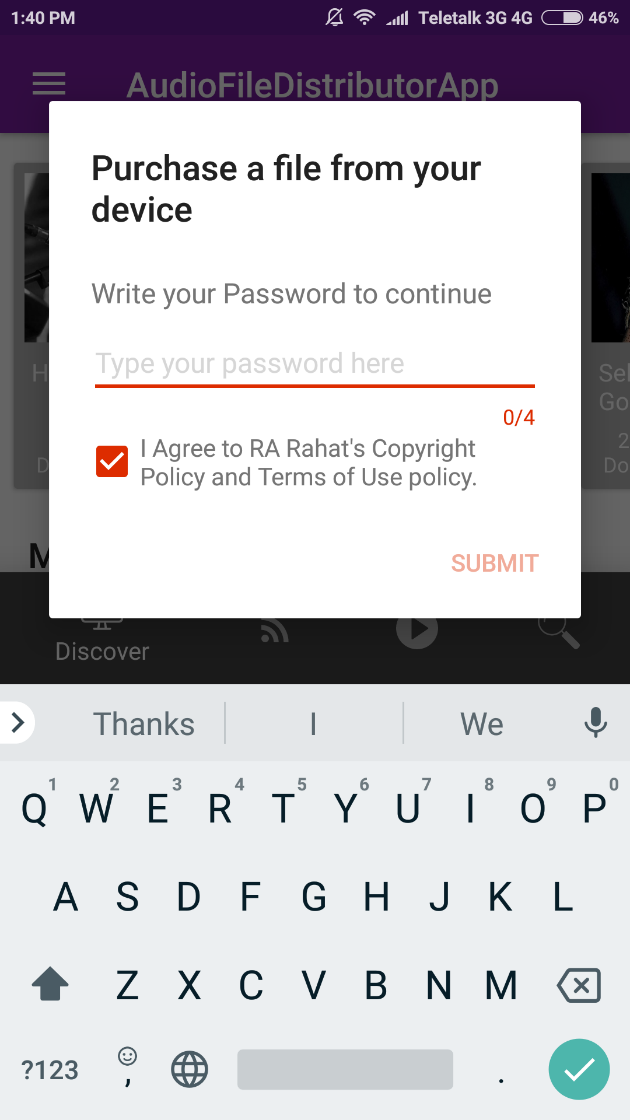
Here the first screenshots represents the “Discover” view and second screenshots represents “Feed” view.

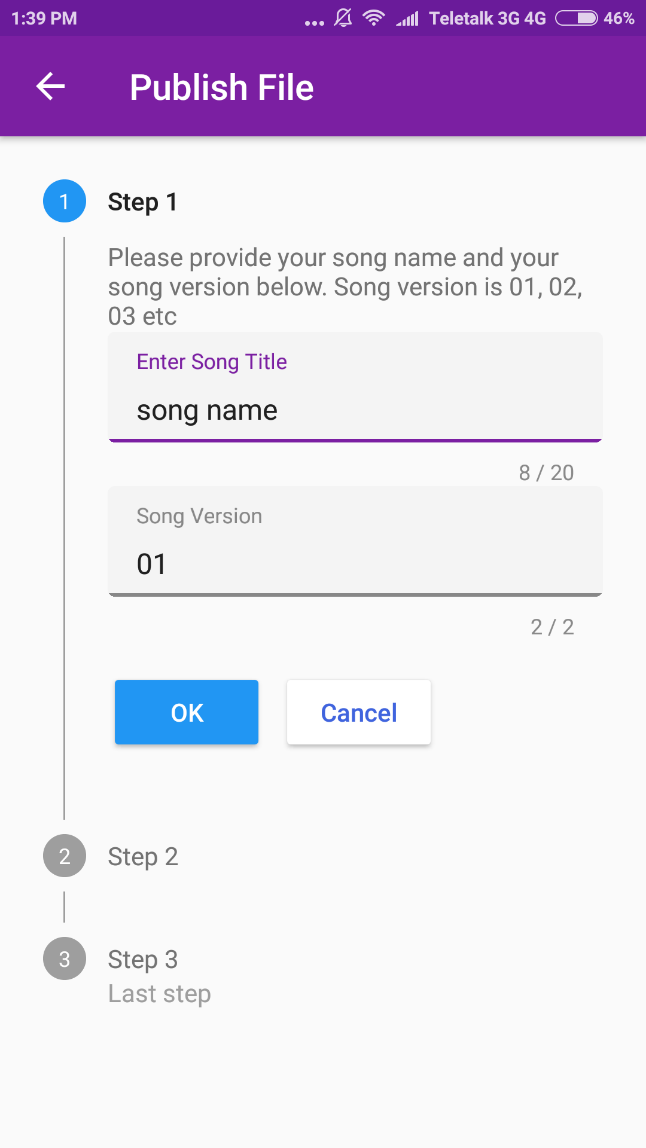
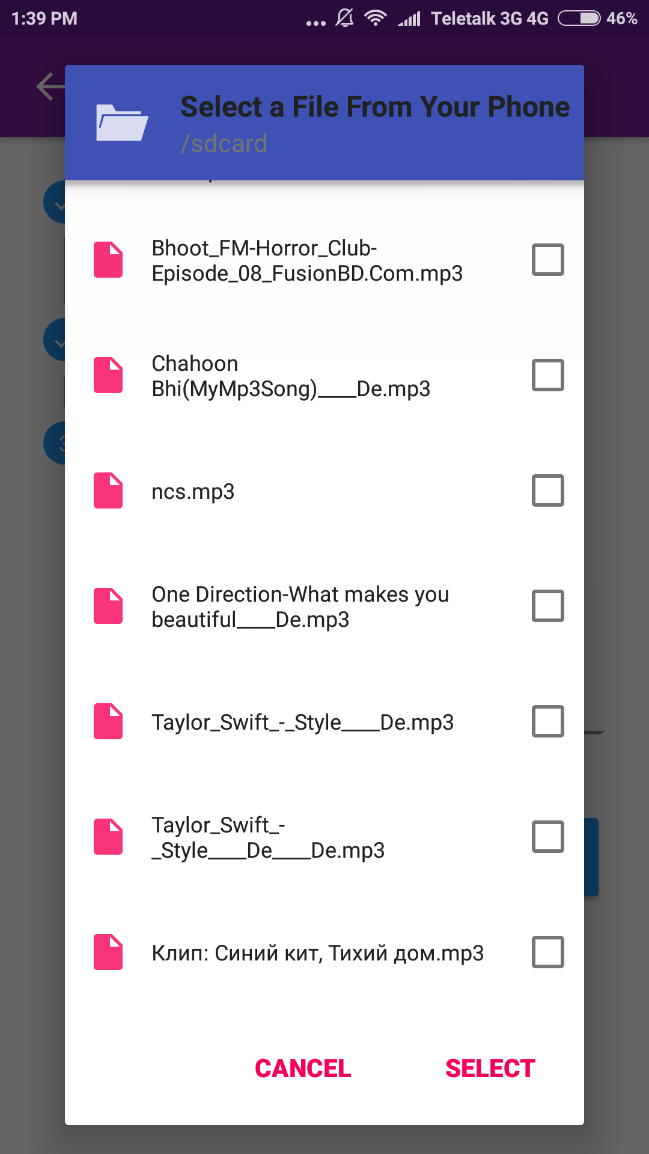
Here the first screenshots represents the “Search” view and second screenshots represents “Music Player” view.

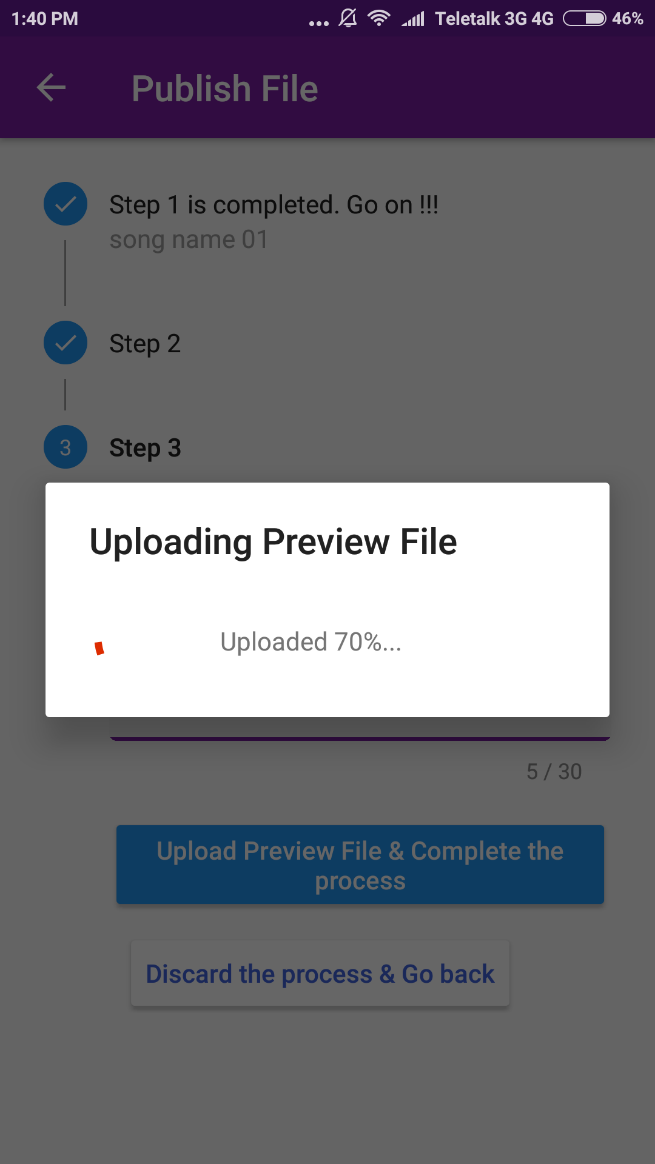
 

Here the first screenshots represents the “Artists More” view and second screenshots represents “Purchase File Download premission” view.

Following four screenshots represents the Artist’s File upload process.

### Chapter 13: Conclusion

I am pleased to submit the project report on **Encryption based Audio File Distribution System With Compatible Music Player** project. From this, the readers will get a clear and easy view of the system. This document can be used effectively to maintain software development cycle in future. It will be very easy to conduct the whole project using this document. I believe that reader will find it in order.