SYMPHONY: As You Like It!

Music Recommendation System

Submitted by:
(101503004) ABHIMANYU SHARMA
(101503023) AMANDEEP SINGH
(101503086) HARNOOR SINGH BEDI
BE Third Year, CSE
Project Team No. CPG-142

Under the mentorship of:
Ms. Vineeta Bassi (Assistant Professor)

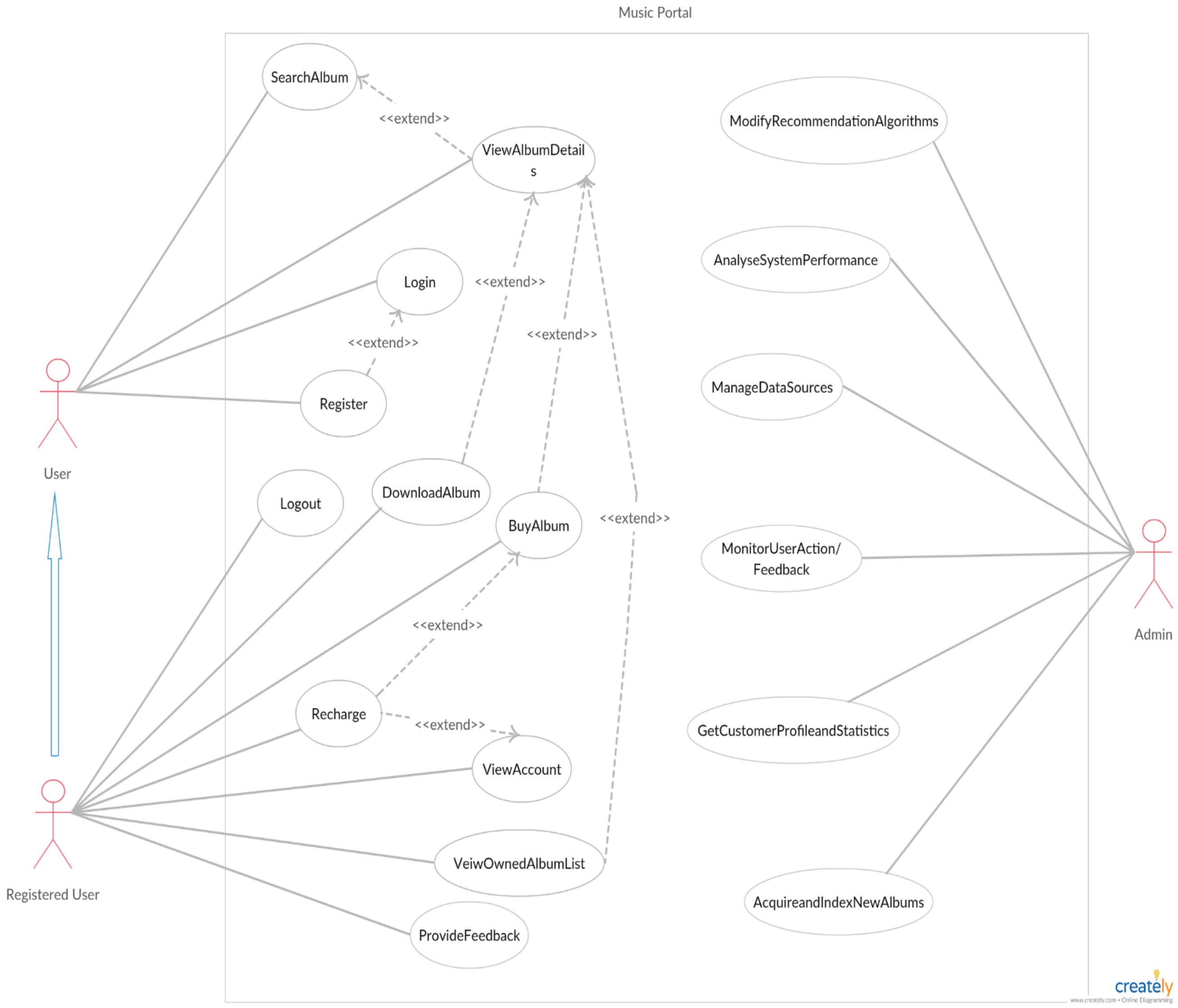
Mr. Sanjeev Rao (Lecturer)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala

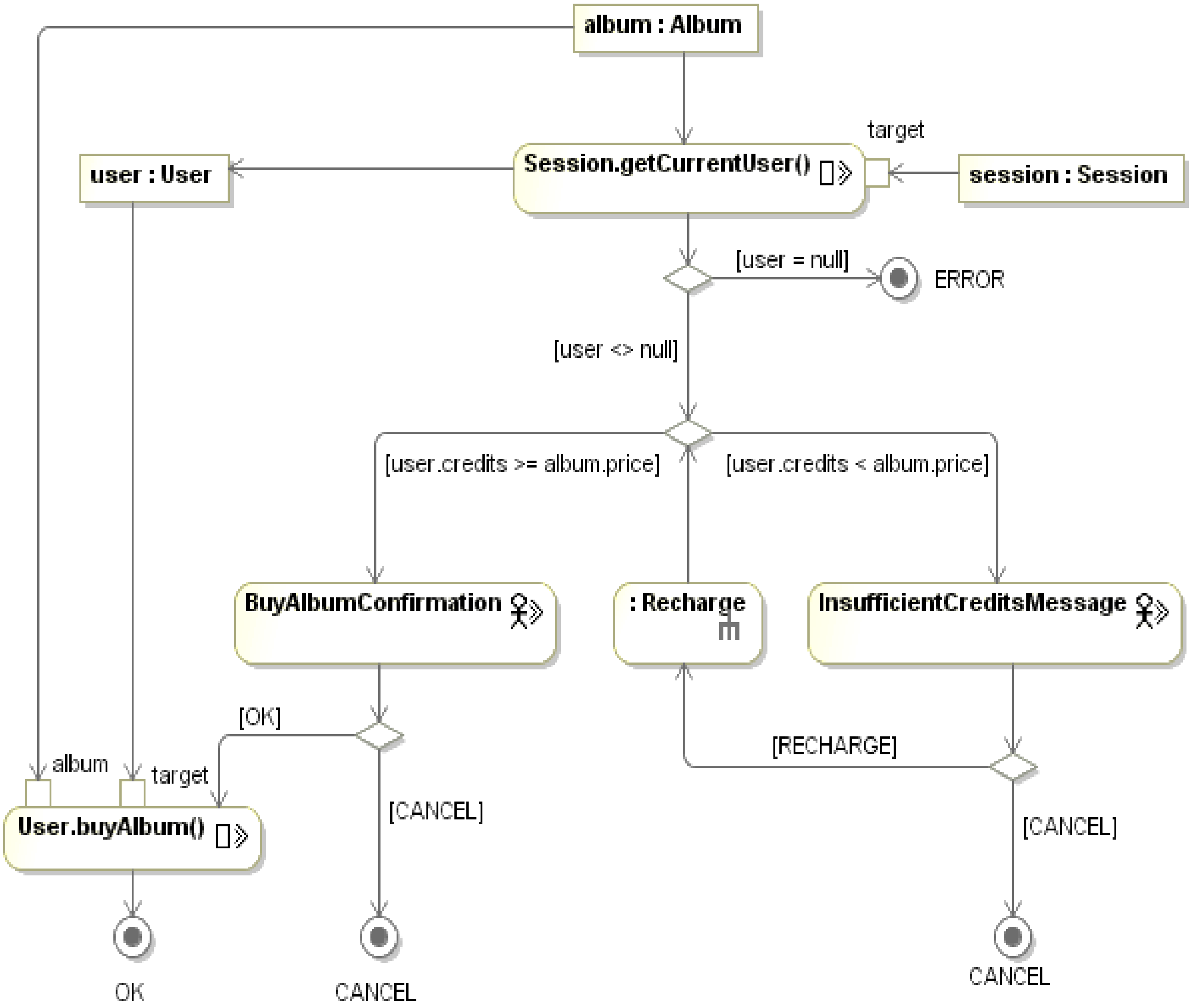
March, 2018

1.Use Case Diagram:

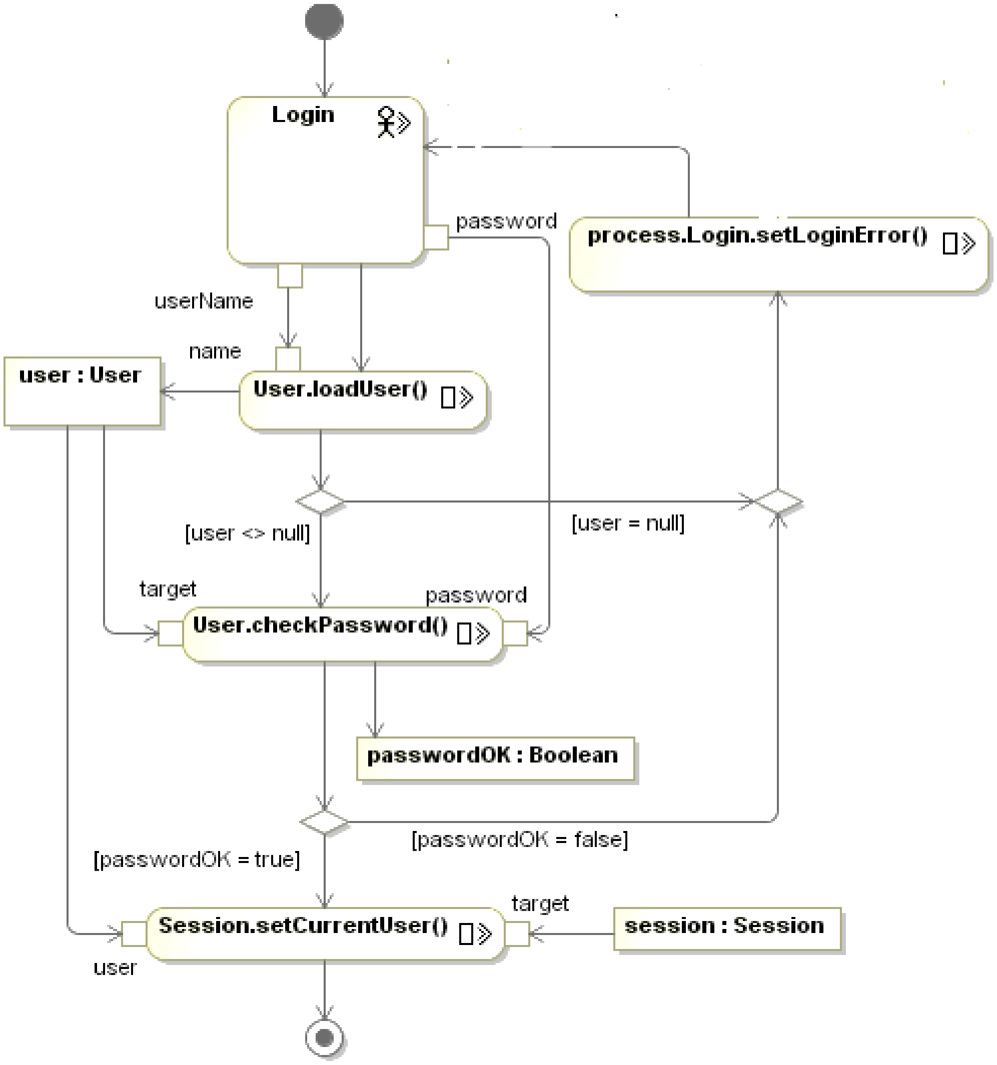


2.Activity Diagram:

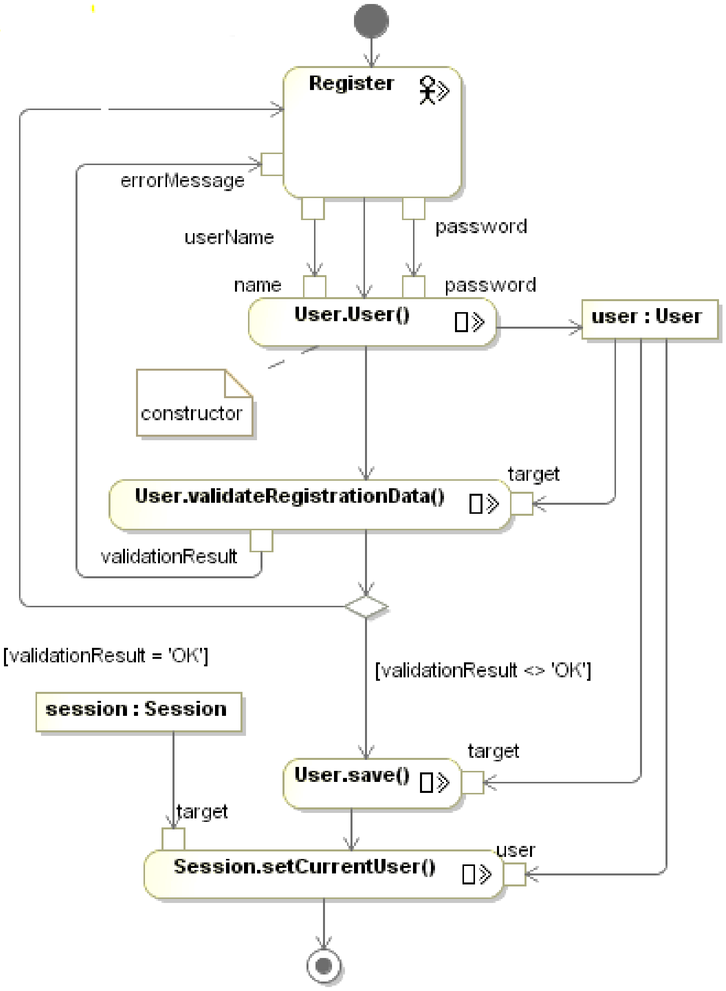
a. Buy Album:



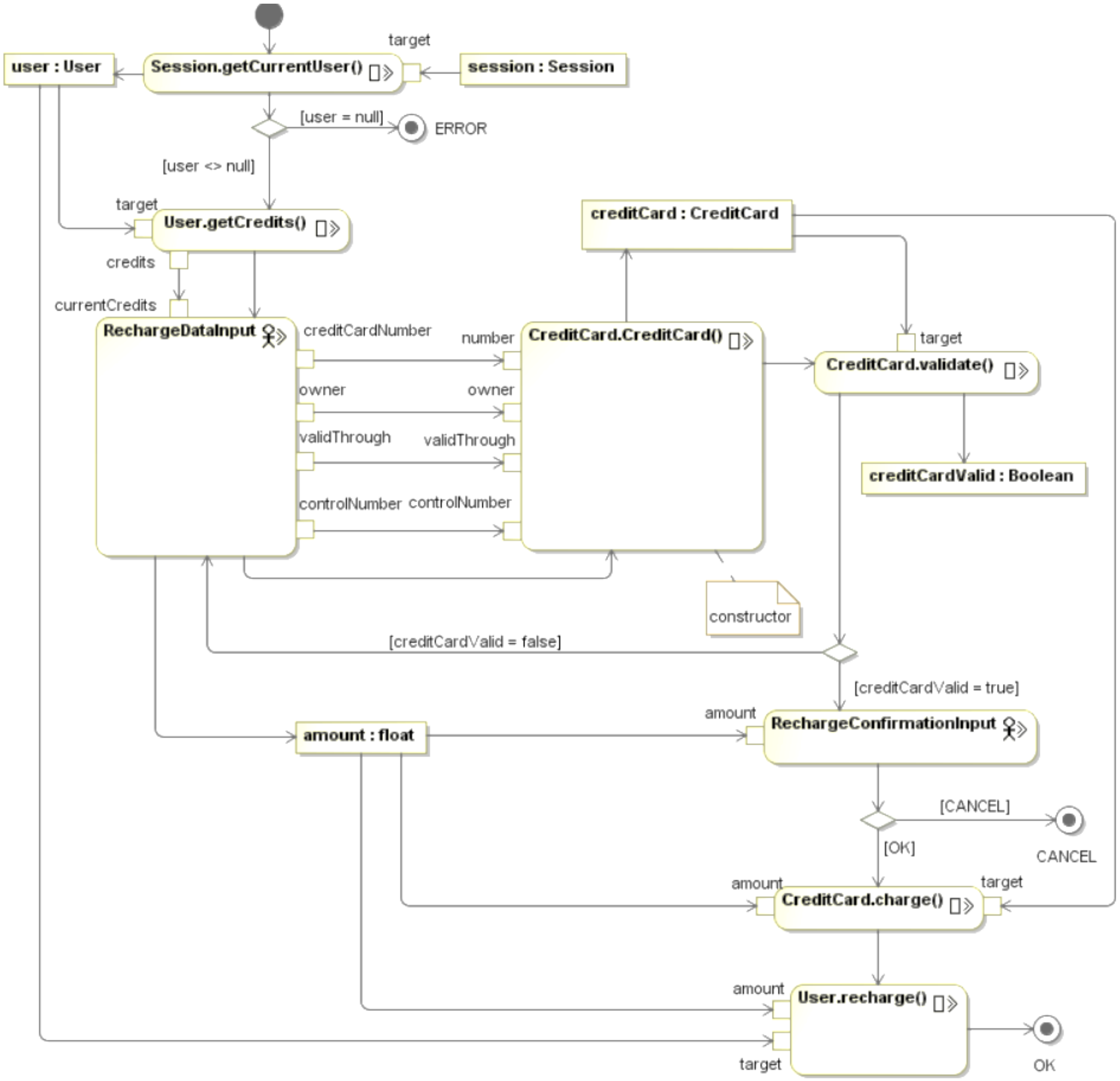
b <u>Login:</u>


Login
password
process.Login.setLoginError()
userName
name
user : User
User.loadUser()
[user <> null]
[user = null]
target
password
User.checkPassword()
passwordOK : Boolean
[passwordOK = true]
[passwordOK = false]
target
Session.setCurrentUser()
session : Session
user

c. Register:

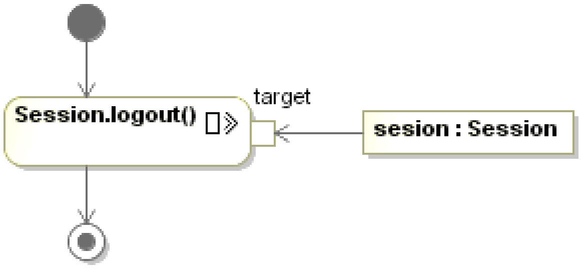

Register
errorMessage
userName
password
name
password
User.User()
user : User
constructor
target
User.validateRegistrationData()
validationResult
[validationResult = 'OK']
[validationResult <> 'OK']
session : Session
target
User.save()
target
Session.setCurrentUser()
user

d. Recharge:

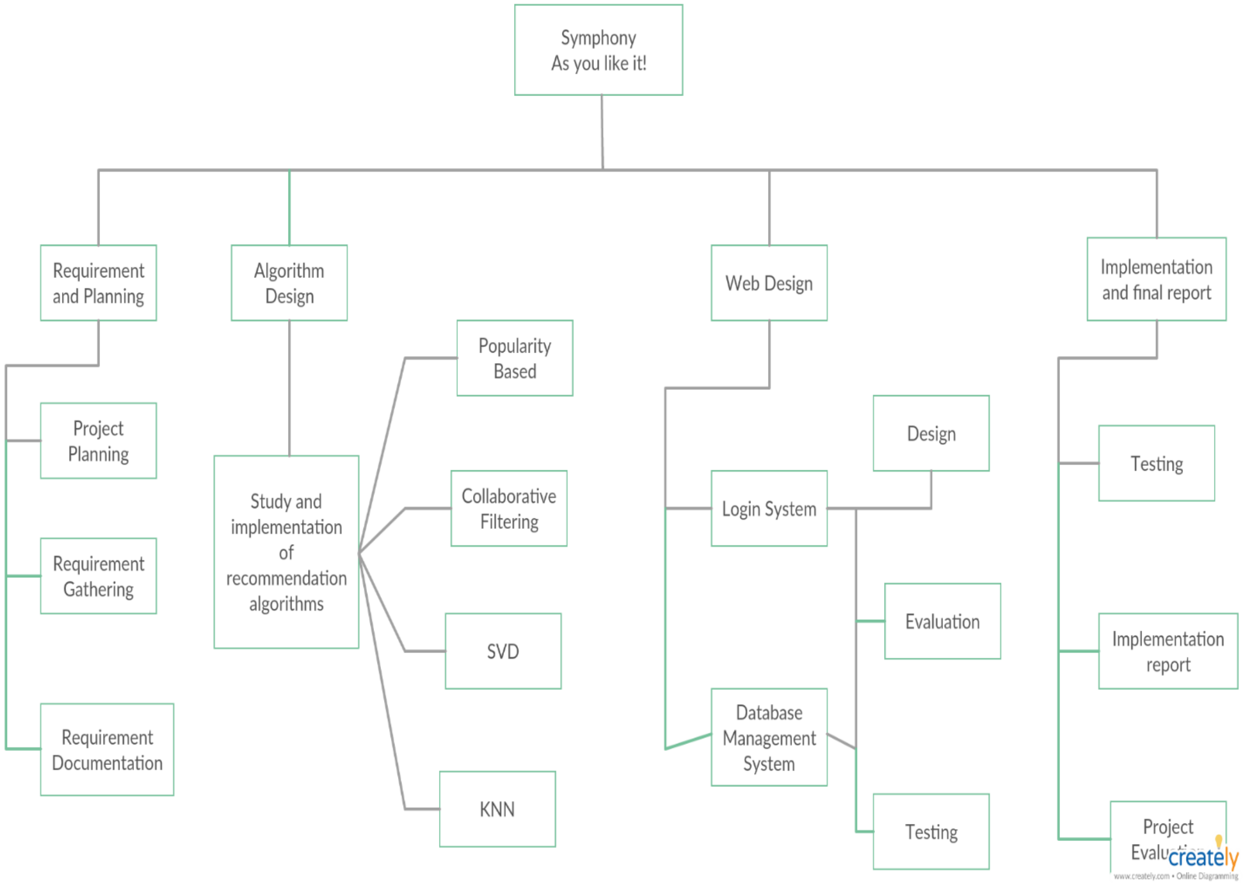


target
user : User
Session.getCurrentUser()
session : Session
[user = null]
ERROR
[user <> null]
target
User.getCredits()
credits
currentCredits
creditCard : CreditCard
RechargeDataInput
creditCardNumber
number
CreditCard.CreditCard()
target
CreditCard.validate()
owner
owner
validThrough
validThrough
creditCardValid : Boolean
controlNumber
controlNumber
constructor
[creditCardValid = false]
[creditCardValid = true]
amount
RechargeConfirmationInput
amount : float
[CANCEL]
[OK]
CANCEL
amount
target
CreditCard.charge()
amount
User.recharge()
target
OK

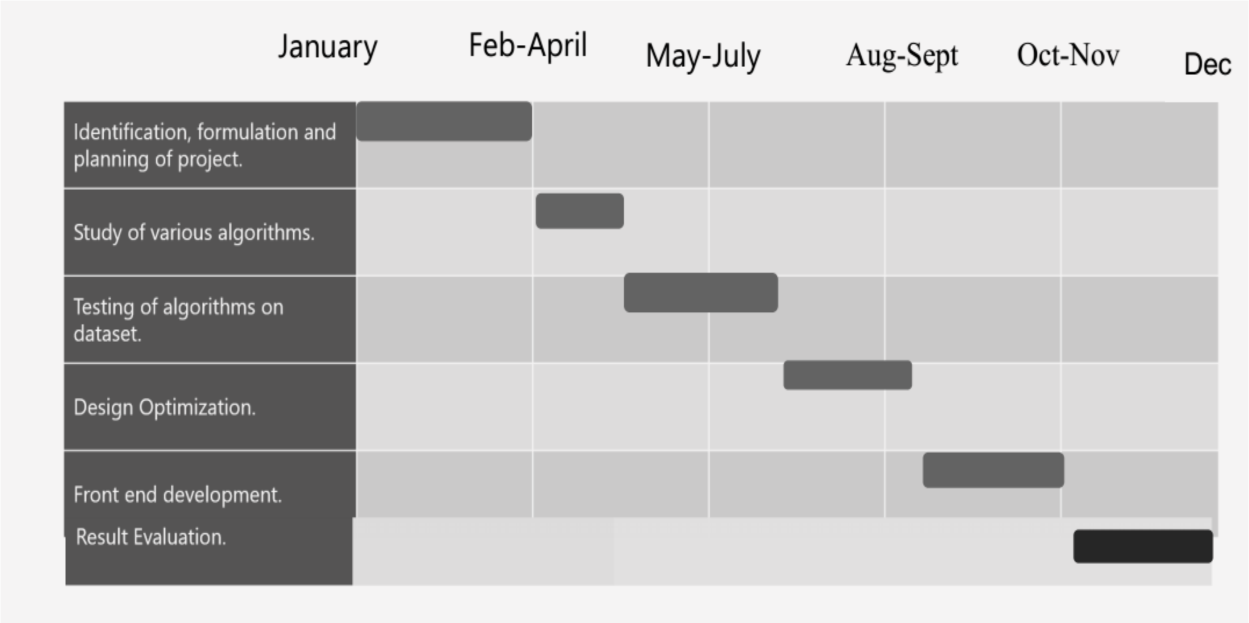
e. Logout:



3. WBS (Work Breakdown Structure):



4. WBS using Gantt chart:



5. Functional and Non-Functional requirements:

a. Functional Requirements:

a. Client: The client-side of the system will be an application with a user interface that is integrated into a music listening website or application. This application gathers the information from users, investigates some actions of the users, and provides the connection with the server. This application is the client-side interface of the Music Recommender, so it does not include the functionalities of the host music environment such as playing music etc.

● Requesting recommendations: The client-side application must allow user to request recommendations manually and interact with the server to receive recommendations.

● Display recommendations: The application must display the recommendations that are obtained from the server to the user in a proper way by providing a GUI.

b. Server: The server-side system will hold the entire data in a graph database and must include all functionality to perform operations on this database, receive requests from the clients, evaluate, create and send recommendations etc.

● Handle recommendation requests

The server application shall obtain and handle requests for recommendations.

- Store evaluations

The server application shall receive and store music evaluations

- Data updating

The server application shall be able to store the newly retrieved data to the database.

- Generate Recommendations

The server application can produce recommendations by interpreting the content and evaluations by actual user. Server will have a recommendation class which contains functions of algorithm.

b. Non-Functional Requirements:

a. Performance requirements:

- Accuracy

Since we will give the priority to the accuracy of the software, the performance of the Music Recommender will be based on its accuracy on recommendations.

● Speed

The system should generate and provide personalized recommendations to the users in a reasonable time.

b. Design constraints:

● Hardware Constraints

The system will be integrated with a website. To use recommendation system, user should enter from a personal computer, mobile device with internet connection, tablet etc.

● Software System Attributes

➔ Usability The software will be embedded in a website. It should be scalable designed to be easily adopted by a system.

➔ Reliability The system should have accurate results and a fast response to user's changing habits. It also should work properly for a reasonably long amount of time. The probability of self-induced failure must be low.

➔ Security User profile information will be used, so data security is one of the most important concerns of the system

6. **Tasks**:

a. Requirement and Planning:

The initial task in the project involves gathering all the requirements and devising a plan to integrate all the requirements into the final product. The plan involves the distribution of workloads among the team members according to their strengths.

b. Algorithm Design:

This task involves the implementation of various algorithms required for the recommender system integrated into the music portal. We will aim to further optimize the algorithms by measuring certain performance metrics.

c. Web Development:

This task involves the design of a user-friendly interface at the front end and the database of albums and artists at the backend.

d. Implementation and Final Report:

This task involves the complete implementation of the project by integrating all the individual components. It also involves documentation and a final report of the project.

7. Use Case Template:

Use Case Name	**SearchAlbum**
Super Use Case	NA
Actor	User
Brief Description	When user will search for an album, then albums in database corresponding query will be shown to the user.
Preconditions	The registered user must be logged in.
Post-Conditions	User will get his/her searched result on the website.

Use Case Name	**ViewAlbumDetails**
Super Use Case	SearchAlbum
Actor	User
Brief Description	Details corresponding to the typed query will be shown to the user.
Preconditions	User must have searched for the album.
Post-Conditions	Details will be shown about the corresponding album.

Use Case Name	**Login**
Super Use Case	NA
Actor	User
Brief Description	When the user logs in to the website; Music Recommendation System will be informed and a recommendation session for the user will start and generate recommendations.
Preconditions	User must be registered.
Post-Conditions	User will see home screen of his profile along with some recommended music.

Use Case Name	**Register**
Super Use Case	NA
Actor	User
Brief Description	When the user creates a new account on the website/application that the recommendation system runs on, also implicitly an account for the user inside the recommendation system will be created to store information about the user profile and statistics to provide better recommendations.
Preconditions	NA
Post-Conditions	User will become a registered member of the website.

Use Case Name	**Logout**
Super Use Case	NA
Actor	RegisteredUser
Brief Description	When the user logs out from the website; Music Recommendation System will be informed and the associated recommendation session for the user will be closed.
Preconditions	User must have logged in.
Post-Conditions	NA

Use Case Name	**DownloadAlbum**
Super Use Case	ViewAlbumDetails
Actor	RegisteredUser
Brief Description	User can download his/her bought albums.
Preconditions	User must be registered and must have bought the album.
Post-Conditions	Album will be downloaded to the user’s local machine.

Use Case Name	BuyAlbum
Super Use Case	ViewAlbumDetails
Actor	RegisteredUser
Brief Description	User can buy his/her favorite album.
Preconditions	User must have enough credit to buy the album.
Post-Conditions	Album will be available for listening and download.

Use Case Name	Recharge
Super Use Case	BuyAlbum, ViewAccount
Actor	RegisteredUser
Brief Description	User can add credit to his account.
Preconditions	User must be a registered user and have valid credit card.
Post-Conditions	Account will have recharged with credited amount.

Use Case Name	ViewAccount
Super Use Case	NA
Actor	RegisteredUser
Brief Description	User can view his account details. User can check or edit his/her playlists.
Preconditions	User must be a registered user.
Post-Conditions	Users will be able to see his account details.

Use Case Name	ViewOwnedAlbumList
Super Use Case	ViewAlbumDetails
Actor	RegisteredUser
Brief Description	User can view his/her bought albums.
Preconditions	User must be a registered user.
Post-Conditions	Owned album of the user will be shown and will be available to download.

Use Case Name	**ProvideFeedback**
Super Use Case	NA
Actor	RegisteredUser
Brief Description	User can rate songs or albums.
Preconditions	User must be a registered user.
Post-Conditions	Ratings will be given to song or album.

Use Case Name	**ModifyRecommendedAlgorithms**
Super Use Case	NA
Actor	Admin
Brief Description	Admin can change the algorithms according precision, recall and accuracy
Preconditions	Not applicable
Post-Conditions	Better accuracy can be obtained.

Use Case Name	**AnalyzeSystemPerformance**
Super Use Case	NA
Actor	Admin
Brief Description	Precision, recall and accuracy can be monitored.
Preconditions	NA.
Post-Conditions	Whole system performance is well monitored.

Use Case Name	**ManageDataSources**
Super Use Case	NA
Actor	Admin
Brief Description	Database and dataset is managed.
Preconditions	NA.
Post-Conditions	New entries are added in database making it more versatile.

Use Case Name	**MonitorUserActions/Feedback**
Super Use Case	NA
Actor	Admin
Brief Description	Feedback of user is monitored and applied using algorithms.
Preconditions	NA.
Post-Conditions	These ratings can be applied through algorithms to improve recommendation system.

Use Case Name	**GetCustomerProfileandStatistics**
Super Use Case	NA
Actor	Admin
Brief Description	Admin can check customer profiles and can obtain statistics.
Preconditions	NA
Post-Conditions	Admin can get insight of what user listens to most.

Use Case Name	**AcquireandIndexNewAlbums.**
Super Use Case	NA
Actor	Admin
Brief Description	Admin can add more albums into database.
Preconditions	NA
Post-Conditions	Database is expanded, and more songs are added to database.