
Multimedia Content Database Management (CSC 675-775-03)

Drashti Pareshkumar Shah
SFSU ID: 920050609
Github Username: DrashtiShah23

Milestone 1

Milestone/Version	Date
M2V1	4/17/2022
M1V1	3/11/2022

Table of Contents

1. Project Description	3
2. Use Cases:	4
3. Database Requirements	8
4. Detailed List of Main Entities, Attributes And Keys	12
5. Entity Relationship Diagram(ERD)	17
6. Testing Table	18
7. Database Model/EER	22
8. Testing SQLQuery	30

Section I: Project Description

In a world with stellar speed internet connection, our way of listening to music has changed drastically. Earlier we had cassettes and music albums to listen to music. We find ourselves amidst cultural diversity, so why not explore different music, podcasts and audiobooks from all over the world? Imagine a convenient way of streaming not only music, but also podcasts and audiobooks in a matter of clicks. This Multimedia Content Database System provides an option to stream music, podcasts and audiobooks on the fly so each kind of user can enjoy the content that they can like. The user can create an account and login and start enjoying the content that they like. The database system offers two types of accounts: Basic and Premium. Basic account users can enjoy music and podcasts free of cost. Even though users could be short on money, with this system, users will never be short of free content. However, Basic accounts contain multimedia with advertisements which some people may find interruptive. Our database system provides a solution by offering a Premium account where users can pay a small fee and enjoy ad - free content. In premium accounts we also offer the users to stream unlimited audioBooks along with music and podcasts. The database system also gives an option to premium users to download the content if they are in a situation with a bad network. The database system also provides users with several discounts. The database system has a premium plan for students where they pay 50% of the premium price and still be able to enjoy all the premium features. They just need to register using their school email id. Also there's a group plan which allows users to use their same account but only one user takes care of the payment instead of everyone paying individually for a premium account. This way the group plan users can still enjoy premium features at a slightly reduced price.

The database system is designed to fulfill the needs of all different kinds of users and provide easily accessible content across the globe.

Section II : Use Cases

1. **Use Case:** Difficulty finding free music.

Actor: Jacob(User), basic account, music, ads

Description: Jacob is a factory worker. He finds his job to be absolutely mundane and he's looking for an option to listen to free music while working. However, he is a cheapskate. He doesn't like spending money until it is absolutely necessary. He doesn't want to waste his time finding good music on the internet and downloading each song he likes. He doesn't mind listening to ads, he just wants a service which he can use for free and enjoy his time at work.

The Multimedia Content Database Management System is there to help Jacob by providing him with free music with the basic plan. He'll just need to register. Moreover, he will not have to spend money and have a pleasant experience while he's working or whenever he wishes to listen to music.

2. **Use Case:** Lack of downloadable/offline content

Actor: Adi(Businessman), User, payment, download multimedia, premium account

Description: Adi is a business owner who frequently flies for his business meetings. Being a businessman, he values his time tremendously. He always wants to stay informed and up to date with the ongoing events in the world. So, while being on his flight, he wants to listen to news and current events podcasts. However, a lot of flights don't provide an option for paid wifi and so Adi is looking for an option where he can download the news podcasts before boarding his flight.

The Multimedia Content Database Management System is there to help Adi as the database system provides a premium user to download multimedia content. The database will therefore help and provide Adi with a platform where he can download the video podcasts of his choice and spend his time productively.

3. **Use Case:** Lack of affordable prices for streaming music for students

Actor: Adele(User), payment, student plan, premium account, podcasts, music

Description: Adele is a college student. She is a part time worker and a full time student. She loves to listen to podcasts or music in her free time. She also listens to calming music while studying. Being a college student, she doesn't want to spend money to listen to music or podcasts. She's now looking for a solution to obtain affordable plans to listen to podcasts and stream music all in one place.

The Multimedia Content Database Management System is there to help Adele by providing her an option to choose the "Student Premium Plan" . With the plan she will be charged only 50% of the premium membership after she registers with her school email id. In this way she'll be able to enjoy the premium features and save some money in the process.

4. **Use Case:** Need for group plan for multiple members

Actor: Mary(Admin), Bob(User), Lincoln(User), Iva(User), group plan, premium account, payments

Description: Mary is the youngest cousin in her extended family. All of her cousins use a premium plan to stream music, podcasts or AudioBooks. She finds that there are no group plans where all of them can have an option where they could enjoy premium features. She notices that they all spend about 15 dollars a month individually. So she's in search of a plan where it is a little bit cheaper for all of them and **one** person can pay for all of them.

The Multimedia Content Database Management System is there to help Mary by providing her an option to choose the “Group Premium Plan” where one member can conveniently take care of the payments and rest of the users can enjoy the premium features. This way Mary, Bob, Lincoln and Iva can enjoy the content and one of the users becomes the admin who takes care of the payment.

5. **Use Case:** Lack of AudioBooks

Actor: Gina(User), audiobooks, premium account

Description: Gina is a lawyer. She was an avid reader growing up. However as time passed by, her life became extremely busy. She doesn't have time to sit-back and read a paperback with her busy lifestyle. She wants to be able to be “reading” books absolutely everywhere she goes. She is looking for an option where she can listen to audiobooks as she wants to make her life more productive, encounter more diverse ways in which she sees the world, and gain pure enjoyment that comes from learning about topics of interest without going through the physical process of reading.

The Multimedia Content Database Management System is there to help Gina by providing her a premium plan where she can not only enjoy ad - free music and podcasts but also audiobooks. Generally the user needs to buy individual audiobooks everytime, but this database management system charges a flat fee every month providing the user with a lot of options. She will even have the feature to download the audiobooks if she ever finds herself in a situation without a network.

6. **Use Case:** Difficulty resuming music any time user wants

Actor: Julia(User), basic account, music, audiobooks, Multimedia tracker

Description: Julia wants to play a song from the Multimedia Content Database System, the system asks Julia to register first as a prerequisite to listen to the song. Marie registers into the system with a basic account, and then logs in her account. Then, she can search for the song she wants to listen to. Marie finds the song and starts playing it. However she quits the

application for attending her phone call and loses the track and time stamp of the music and the song starts playing from the beginning.

The Multimedia Content Database Management System is there to help Julia. The system has a Multimedia Tracker which allows her to pause and play the track how much ever time she wants and it wouldn't always start from the beginning if she quits the application.

SECTION III: Database Requirements

1. General User
 - a. A general user is a registered user, or admin.
 - b. A general user shall be able to create many accounts with different emails.
 - c. A general user shall be able to register to many mailing lists
2. Roles
 - a. A role shall be used by many accounts.
3. Admin
 - a. An admin is a registered user.
 - b. An admin can edit the roles.
 - c. An admin can add new roles.
 - d. An admin can delete roles.
4. Payment Method
 - a. A payment method is a bank account, paypal or credit card.
 - b. A bank account payment method has a debit card.
 - c. A payment method can be linked to many premium accounts.
5. Registered users
 - a. A registered user is a general user.
 - b. A registered user can login into his/her account from one device.
 - c. A registered user shall do at least one IP address validation every time he/she changes visitor networks.
 - d. A registered user shall be able to create only one free basic account
 - e. A registered user shall be able to create only one premium account
 - f. A registered user with a basic account subscription has access to only music and podcasts but not AudioBooks.
 - g. A registered user with a basic account subscription will encounter ads in the multimedia content.
 - h. A registered user with a premium subscription has access to all the multimedia content.

- i. A registered user with a premium subscription will not encounter ads in the multimedia content.

6. Account

- a. An account is either a basic or premium account.
- b. An account shall belong to only one user.
- c. An account shall be located in only one region.
- d. An account can have many roles.

7. Basic Account

- a. A basic account is free for registered users.
- b. A basic account is an account.
- c. A basic account shall have permissions to stream only music and audio podcasts.
- d. A basic account shall have ads in many multimedia content.
- e. A basic account shall have many users.

8. Premium Account

- a. A premium account is a paid account.
- b. A premium account is an account
- c. A premium account shall have an expiration date.
- d. A premium account shall have many users.
- e. A premium account shall have individual plan, student plan and group plan.
- f. A premium account shall have permissions to stream unlimited audio podcasts, music and Audiobooks multimedia content.
- g. A premium account shall have permissions to download unlimited audio podcasts, music and Audiobooks multimedia content.
- h. A premium account shall have at least one preferred payment method.

9. Individual Premium Plan

- a. An individual premium plan has all the features and permissions of the premium account.
- b. An individual premium plan shall have a `individualplan_id`

10. Premium Student Plan

- a. A premium student plan has all the features and permissions of the premium account.
- b. A premium student plan shall have a 50% discount on the price of the premium plan.
- c. A premium student plan shall have a `student_plan_id`

11. Premium Group Plan
 - a. A premium group plan has all the features and permissions of the premium account.
 - b. A premium group plan will have at most 4 profiles managed and controlled by the admin.
 - c. A premium group plan shall have a group_plan_id
12. Multimedia Content
 - a. A multimedia content is either an audio podcast, music, or AudioBook
 - b. A multimedia content shall be either original or an exact copy of itself
 - c. A multimedia content shall have many authors
 - d. A multimedia content can have ads.
 - e. A multimedia content shall be located in at least one Content Delivery Network.
13. Audio Podcasts
 - a. Audio multimedia content shall be streamed by many accounts
 - b. Audio multimedia content shall be streamed by many premium accounts.
 - c. Audio multimedia content shall be downloaded by many premium accounts.
 - d. Audio multimedia content shall be able to be resumed as many times as needed.
14. Music
 - a. Music multimedia content shall be streamed by many accounts
 - b. Music multimedia content shall be streamed by many premium accounts
 - c. Music multimedia content shall be downloaded by many premium accounts
 - d. Music multimedia content shall be able to be resumed as many times as needed.
15. AudioBook
 - a. An AudioBook shall be streamed only by premium accounts.
 - b. An AudioBook shall be downloaded only by many premium accounts.
 - c. An AudioBook shall have many publishers.
16. Content Delivery Network (CDN)
 - a. A CDN shall be located in only one region
 - b. A CDN contains many copies of the original multimedia content.
 - c. A CDN shall be represented in at least one manifest.
17. Manifest Component

- a. A manifest shall have the ID's of many multimedia contents (copies)
- b. A manifest shall have the location (CDN) of many multimedia contents (copies).
- c. A manifest shall be located in the main cloud server (e.g AWS).

18. Region

- a. A region shall have many accounts.
- b. A region shall have many CDN.

19. Profile

- a. A profile shall be owned by only and only one premium group account.
- b. A profile shall have one alias.

20. Advertisements

- a. All advertisements shall have many organizations
- b. All advertisements shall have advertisement id.
- c. An advertisement can be linked to many multimedia contents.

21. Organizations

- a. All organizations shall have a unique organization id.
- b. All organizations shall have many ads.

22. Promotions:

- a. All promotions shall have a unique promotion id.
- b. All promotions shall have many ads.
- c. Promotions shall have promotion type: self or partnered.

SECTION IV: Detailed List of Main Entities, Attributes and Keys

1. General User (Strong)

- user_id: key, numeric
- name: composite, alphanumeric
- email: alphanumeric
- password: alphanumeric

2. Mailing List (Strong)

- email: alphanumeric
- Mailing_id: key, alphanumeric (keep track of mails the system sends)
- description: alphanumeric

3. Devices (Strong)

- device_id: key, alphanumeric
- Device_ip: key, alphanumeric
- Account_id: key, alphanumeric

4. Role (Strong)

- role_id: key, numeric
- description: alphanumeric
- permission_number: numeric
- admin_id: key, alphanumeric

5. Authors(Strong)

- name: composite, alphanumeric
- description: alphanumeric
- author_id: key, alphanumeric

6. Payment Method(Strong)

- Payment_type: alphanumeric
- Payment_id: key, numeric
- Account_id: key, numeric
- Payment_amount: numeric

7. Multimedia Content(Strong)

- cdn_id: key, alphanumeric
- Premium actions: alphanumeric

- author_id: key, alphanumeric

8. Advertisements(Strong)

- Advertisement_id: key, alphanumeric
- Description: alphanumeric
- Ad_name: alphanumeric, composite

9. Organization(Strong)

- Organization_id: key, alphanumeric
- Description: alphanumeric
- Organization name: composite, alphanumeric
- Organization contact no: numeric, multivalued

10. Publishers(Strong)

- Publisher_name: alphanumeric, composite
- Pushlished_id: key, alphanumeric
- Description: alphanumeric

11. Multimedia tracker(Strong)

- Tracker_id: key, alphanumeric
- Author_id: key, alphanumeric
- Date and time: timestamp, multivalued

12. CDN (strong)

- cdn_id: key, alphanumeric
- manifest_id: key, alphanumeric
- region: alphanumeric
- region_id: key, alphanumeric

13. Regions(Strong)

- Region_id: key, alphanumeric
- Device_id: key, alphanumeric
- Country_code: key, alphanumeric
- Cdn_id: key, alphanumeric
- Region name: alphanumeric, composite

14. Actions (Strong)

- Action_id: key, alphanumeric
- premium_action: key, alphanumeric
- Premium Actions: alphanumeric, table

15. Messaging List(Strong)

- phone_number: multivalued, composite, alphanumeric

- description: alphanumeric
- Messaging_id: key, alphanumeric

16. Promotions(Strong)

- Promotion_id: key, numeric
- Promotion_type: alphanumeric
- Promotion_title: alphanumeric
- Description: alphanumeric

17. Account (Weak)

- account_id: key, numeric
- user_email: key, alphanumeric
- role: key, numeric
- Expires_on: date
- Region: alphanumeric

18. Admin (Weak)

- Admin_id: key, numeric
- Role: key, numeric
- Name: composite, alphanumeric
- Permission_id: key, numeric

19. Registered User(Weak)

- Reg_user_id: key, numeric
- Role: key, numeric
- Name: composite, alphanumeric

20. Basic Account (Weak)

- Basic_id: key, numeric
- Name: composite, alphanumeric
- Role: key, numeric

21. Premium Account (Weak)

- Premium_id: key, numeric
- Name: composite, alphanumeric
- Role: key, numeric

22. Student Plan (Weak)

- Student_plan_id: key, numeric
- Student_plan_cost: numeric
- Premium_id: key, numeric

23. Individual Plan (Weak)

- Individual_plan_id: key, numeric
- Individual_plan_cost: numeric
- Premium_id: key, numeric

24. Group Plan (Weak)

- Group_plan_id: key, numeric
- Group_plan_cost: numeric
- Premium_id: key, numeric

25. Bank Account (Weak)

- Acct_number: key, numeric
- Bank_name: alphanumeric
- Routing_number: numeric
- Acct_holder_name: alphanumeric

26. Debit Card (Weak)

- Deb_card_number: key, numeric
- Expiration_date: composite, date
- CVV: numeric
- Deb_card_holder_name: alphanumeric
- Payment_number: numeric
- Day_of_transaction: date

27. Credit Card(Weak)

- Credit_card_number: key, numeric
- Expiration_date: composite, date
- CVV: numeric
- Cred_card_holder_name: alphanumeric
- Zip Code: numeric
- Payment_number: numeric
- Day_of_transaction: date

28. Paypal (Weak)

- Paypal_id: key, alphanumeric
- Paypal_payment_number: numeric
- Day_of_transaction: date

29. Podcasts(Weak)

- Podcast_id: key, numeric
- Podcast_name: alphanumeric
- Podcast_author: alphanumeric, composite
- Podcast_release_year: date

30. Music (Weak)

- Music_id: key, numeric
- Music_name: alphanumeric
- Music_author: alphanumeric, composite
- Music_release_year: date

31. AudioBook(Weak)

- AudioBook_id: key, numeric
- AudioBook_name: alphanumeric
- AudioBook_author: alphanumeric, composite
- AudioBook_release_year: date

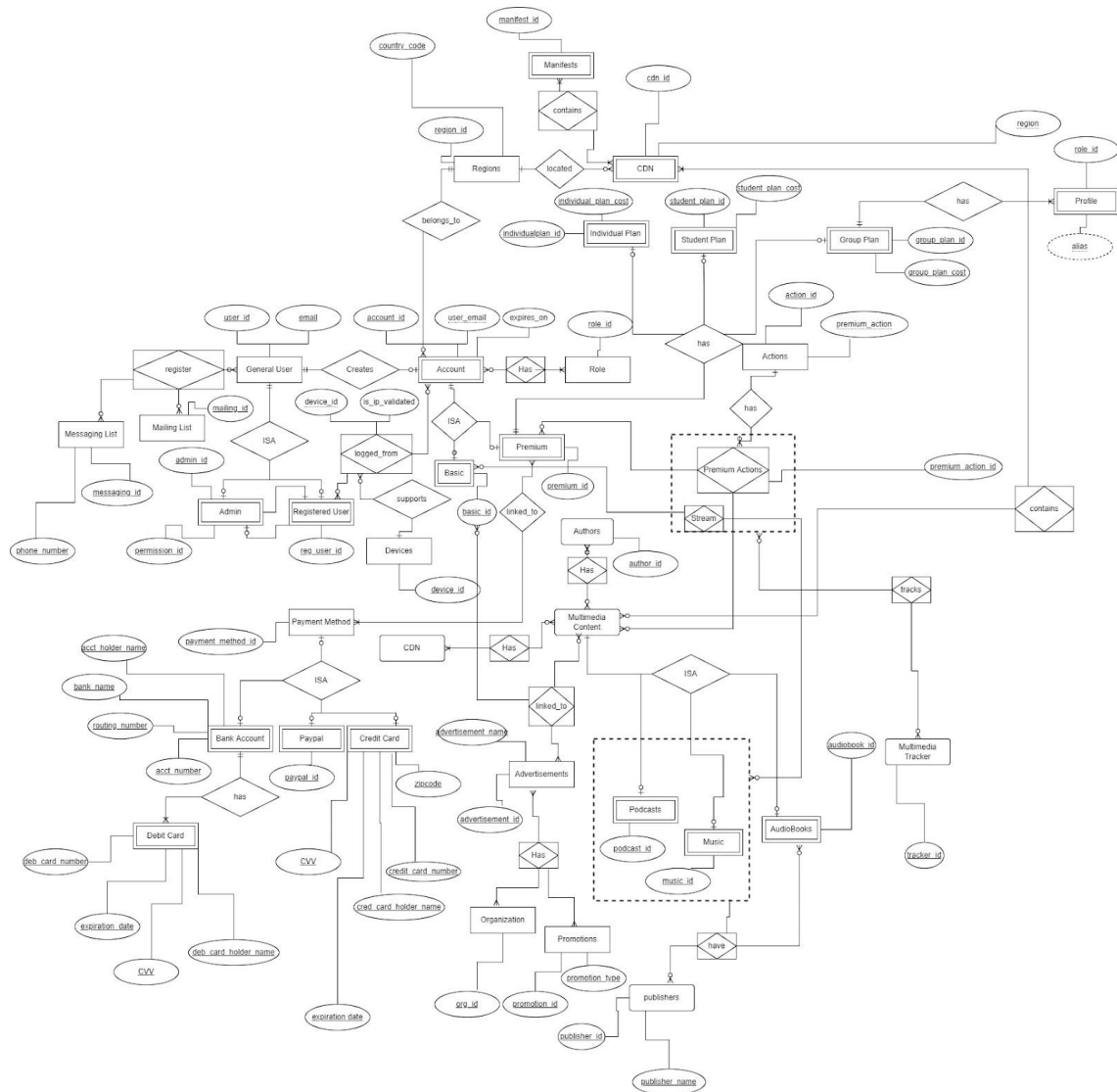
32. Profile(Weak)

- Role_id: key, numeric
- Alias: alphanumeric
- Name: composite, alphanumeric

33. Manifests (Weak)

- Manifest_id: key, numeric
- Manifest_url: alphanumeric
- Region: alphanumeric

Section V: Entity Relationship Diagram



Section VI: Testing Table

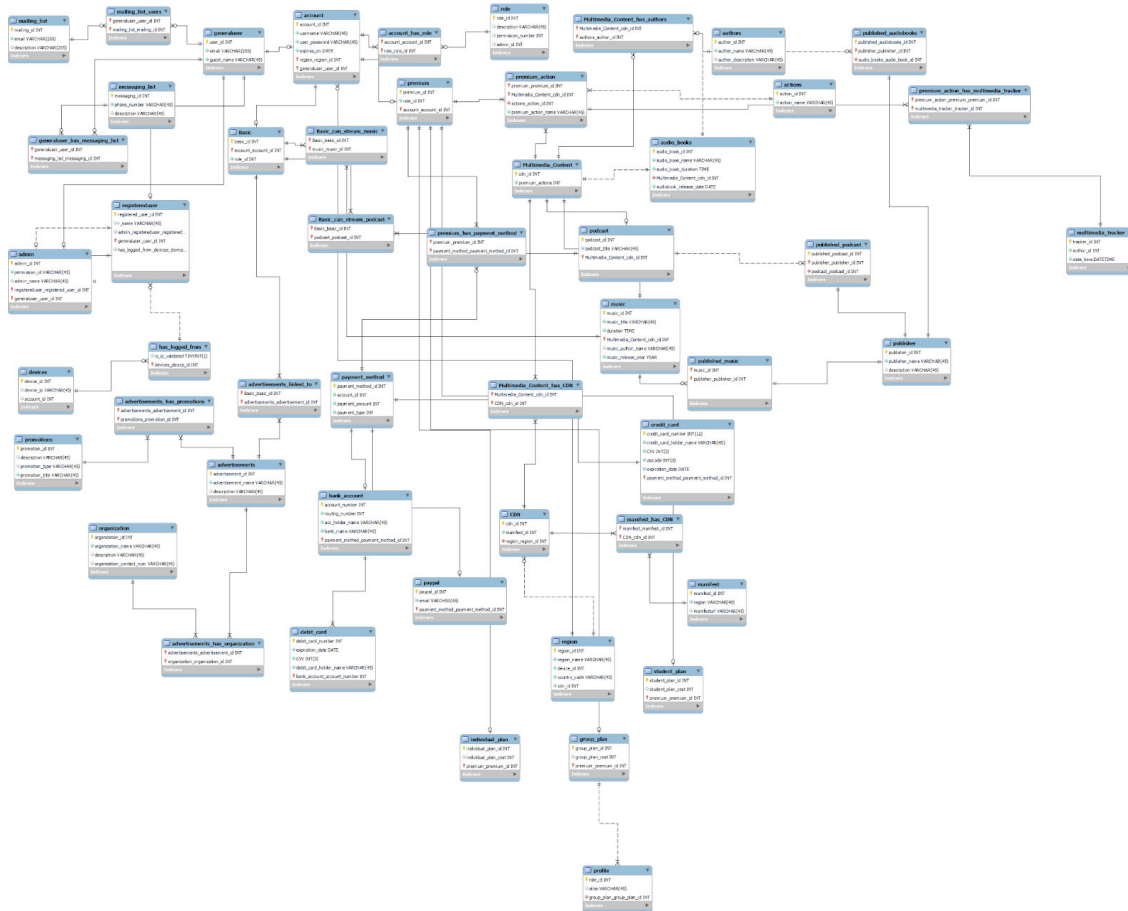
Rule	Entity A	Relation	Entity B	Cardinality	Pass /Fail	Error Description
1.	General User	ISA	Unregistered User	Zero or one	Fail	Users can't be unregistered. They have to be registered or have an admin role.
2.	User	Creates	Account	1 - to -1	Fail	Users can create many accounts with different emails.
3.	User	Registers	Mailing Lists	Zero or many	Pass	None
4.	Admin	Permissions	Role	One or many	Pass	None
5.	Role	Has	Accounts	One or many	Fail	Role can sustain itself even if there's no account. So the cardinality should be zero or many.
6.	Bank Account Payment Method	ISA	Credit Card or Debit Card	Zero or One	Fail	Can't link credit card and debit card together as credit cards can exist even if the user doesn't have a bank account.
7.	Payment Method	ISA	Bank Account / Credit Card/ Paypal	Zero or one	Pass	None
8.	Payment Method	linked_to	Premium Account	Many	Pass	None

9.	User	logged_from	Devices	Many	Fail	Users can login from one Device.
10.	Account	belong	region	One and only one	Pass	None
11.	Account	ISA	Basic Account	Zero or one	Pass	None
12.	Account	ISA	Premium Account	Zero or One	Pass	None
13.	Basic Account	Stream	Music/Podcasts	Zero or many	Pass	None
14.	Basic Account	Has	User	One and only one	Fail	Basic account can have many users
15.	Basic Account	linked_to	Advertisements	Zero or one	Fail	Basic account can be linked to zero or many advertisements
16.	Premium Account	Has	User	One and only one	Fail	Premium Account can have many users.
17.	Premium Actions	Has	Actions	Zero or many	Fail	Premium actions can have only one action but action can have a table(zero or many) of premium actions
18.	Premium Account	Has	Payment Method	Zero or one	Fail	Premium account shall have at least one preferred payment method
19.	Premium Account	Has	Student Plan	Zero or one	Pass	None
20.	Premium Account	Has	Individual Plan	Zero or one	Pass	None
21.	Premium Account	Has	Group Plan	Zero or one	Pass	None
22.	Multimedia Content	Has	Authors	Zero or one	Fail	Multimedia content can have zero or many authors
23.	Multimed	linked_to	Advertis	Zero or	Pass	None

	ia content		ements	many		
24.	Multimedia Content	contains	CDN	One or many	Pass	None
25.	Multimedia Content	contains	Manifest	Zero or one	Fail	Multimedia content can contain one or many manifests
26.	Audio Podcasts	Stream	Premium Accounts	Zero or one	Fail	Audio podcasts can be streamed by many premium accounts
27.	Audio Podcasts	Downloaded	Premium Accounts	Zero or one	Fail	Audio podcasts can be downloaded by many premium accounts
28.	Audio Podcasts	tracked	Multimedia tracker	Zero or many	Pass	None
29.	Music	Stream	Premium Accounts	Zero or one	Fail	Music can be streamed by many premium accounts
30.	Music	downloaded	Premium Accounts	Zero or one	Fail	Music can be downloaded by many premium accounts
31.	Music	tracked	Multimedia tracker	Zero or many	Pass	None
32.	AudioBook	Streamed	Premium Account	Zero or many	Pass	None
33.	AudioBook	Downloaded	Premium Account	Zero or many	Pass	None
34.	AudioBook	have	Publishers	Zero or many	Pass	None
35.	Publisher	Have	Audiobook	One or	Fail	They can have zero

	s		ok	many		or many audiobooks.
36.	CDN	located	region	Zero or one	Fail	CDN can be located in only one region.
37.	Regions	located	CDN	Zero or many	Pass	None
38.	CDN	contains	manifest	One or many	Pass	None
39.	Manifest	has	location	One or many	Pass	None
40.	Region	belongs_to	Account	One or many	Fail	Region can belong to zero or many accounts.
41.	Profile	has	Group Plan	One and only	pass	None
42.	Advertisements	Has	Organizations	Many to many	Pass	None
43.	Advertisements	linked_to	Multimedia Contents	Many to Many	Pass	None
44.	Advertisements	Has	Promotions	Many to Many	Pass	None

Section VII: Database Model/EER



/* I know that the image isn't very clear, but if you zoom in, it looks clearer. Sorry i have more tables and this was the only way I could turn it in*/

A description of all the tables that implement ON DELETE AND ON UPDATE and all their possible constraints (CASCADE, SET NULL...).

TABLE	FK	ON DELETE	ON UPDATE	Comment
mailing_list_users	generaluser_user_id	ON Cascade	ON Cascade	If a user_id is deleted, the mailing list user should also be deleted.
mailing_list_users	mailing_list_mailing_id	ON Cascade	ON Cascade	If a mailing_id is deleted, then they should also be removed from mailing_list_users
generaluser_has_messaging_list	generaluser_user_id	ON Cascade	ON Cascade	If a user_id is deleted, the messaging list user should also be deleted.
generaluser_has_messaging_list	messaging_list_messaging_id	ON Cascade	ON Cascade	If a messaging_id is deleted, then they should also be removed from messaging_list_users
has_logged_from	devices_device_id	ON Cascade	On Cascade	If a device_id is deleted then it should also be deleted from has_logged_on table
registereduser	admin_registereduser_registered_user_id	ON Cascade	ON Cascade	If an admin is deleted, registered user should also be deleted
registereduser	generaluser_user_id	ON Cascade	ON Cascade	If a general user is deleted, registered user should also be deleted
registereduser	has_logged_from_devices_device_id	ON Cascade	ON Cascade	If a device_id is deleted or updated, registered user that was holding the

				device_id will be set to null
admin	registereduser_registered_user_id	ON Cascade	ON Cascade	If a registered user is deleted, then the admin must be deleted too
admin	generaluser_user_id	ON Cascade	ON Cascade	If a general user is deleted, then the admin must be deleted too
account	region_region_id	No action	On Cascade	If a region_id is deleted, the account shouldn't be deleted
Account	generaluser_user_id	On Cascade	On Cascade	If a general user is deleted, the account should also be deleted
account_has_role	account_account_id	On Cascade	On Cascade	If an account_id gets deleted, then the role should be deleted as well.
account_has_role	role_role_id	Restrict	On Cascade	If a role_id is deleted then it shouldn't delete all entries from account_has_role
basic	account_account_id	On Cascade	On Cascade	If an account_id is deleted then the basic account should also be deleted
premium	account_account_id	On Cascade	On Cascade	If an account_id is deleted then the premium account should also be deleted
advertisements_linked_to	advertisements_advertisement_id	On Cascade	On Cascade	If advertisement_id is deleted then wherever it is linked to, it also has to be deleted from that table

advertisements_has_promotions	promotions_promotion_id	On Cascade	On Cascade	If promotions are deleted/updated then the promotions should also be deleted from the advertisements
advertisements_has_organization	advertisement_id	On Cascade	ON Cascade	If advertisement_id is deleted then it also has to be deleted from organization table
advertisements_has_organization	organization_id	On Cascade	ON Cascade	If organization_id is deleted then it also has to be deleted from advertisement table.
premium_has_payment_method	premium_id	On Cascade	On Cascade	If the premium_id is deleted, then the payment method must be deleted
premium_has_payment_method	payment_method_payment_method_id	On cascade	On Cascade	If we delete the payment method_id then it must be deleted from the premium table. Likewise for update
Bank account	payment_method_payment_method_id	On Cascade	On Cascade	If we delete the payment method_id, then it should be deleted from the bank account table
paypal	payment_method_payment_method_id	On Cascade	On Cascade	If we delete the payment method_id, then it should be deleted from the paypal table
credit_card	payment_method_payment_method_id	On Cascade	On Cascade	If we delete the payment method_id, then it should be deleted from the credit_card table
debit_card	bank_account_account_number	On Cascade	On Cascade	If we delete the bank_account_num

	r			ber then it should be deleted from the debit_card table
CDN	region_region_id	On Cascade	On Cascade	If we delete the region_region_id then it should be deleted from the CDN table
Multimedia_Content_has_CDN	CDN_cdn_id	On Cascade	On cascade	If we delete the CDN_cdn_id then it should be deleted from the Multimedia_Content_has_CDN table
Multimedia_Content_has_authors	authors_author_id	On Cascade	On cascade	If we delete the authors_author_id then it should be deleted from the Multimedia_Content_has_authors table
podcast	Multimedia_Content_cdn_id	On Cascade	On cascade	If we delete the Multimedia_Content_cdn_id then it should be deleted from the podcast table
music	Multimedia_Content_cdn_id	On Cascade	On cascade	If we delete the Multimedia_Content_cdn_id then it should be deleted from the music table
audio_books	Multimedia_Content_cdn_id	On Cascade	On cascade	If we delete the Multimedia_Content_cdn_id then it should be deleted from the audio_bookstable
published_podcast	publisher_publisher_id	On Cascade	On cascade	If we delete the publisher_id, then it should be deleted from the podcast table
published_	podcast_podca	On Cascade	On cascade	If we delete the

podcast	st_id			podcast_id, then it should be deleted from the published_podcast table
published_music	podcast_podcast_id	On Cascade	On cascade	If we delete the podcast_id, then it should be deleted from the published_music table
published_music	publisher_publisher_id	On Cascade	On cascade	If we delete the publisher_id, then it should be deleted from the published_music table
published_audiobooks	podcast_podcast_id	On Cascade	On cascade	If we delete the podcast_id, then it should be deleted from the published_audiobooks table
published_audiobooks	publisher_publisher_id	On Cascade	On cascade	If we delete the publisher_id, then it should be deleted from the published_audiobooks table
Basic_can_stream_music	Basic_basic_id	On Cascade	On cascade	If basic_id is deleted then it should be deleted from the Basic_can_stream_music table
Basic_can_stream_music	music_music_id	On Cascade	On cascade	If music_music_id is deleted then it should be deleted from the Basic_can_stream_music table
Basic_can_stream_podcast	Basic_basic_id	On Cascade	On cascade	If basic_id is deleted then it should be deleted from the

				Basic_can_stream_podcast table
Basic_can_stream_podcast	music_music_id	On Cascade	On cascade	If music_music_id is deleted then it should be deleted from the Basic_can_stream_podcast table
premium_action	premium_premium_id	On Cascade	On cascade	If premium_id is deleted then it should be deleted from premium_action table
premium_action	Multimedia_Content_cdn_id	On Cascade	On cascade	If Multimedia_Content_cdn_id is deleted then it should be deleted from premium_action table
premium_action	actions_action_id	RESTRICT	On Cascade	It shouldn't allow to delete the actions_id
premium_action_has_multimedia_tracker	premium_action_premium_id	On Cascade	On cascade	If premium_action_premium_id is deleted then it should be deleted from premium_action_has_multimedia_tracker table
premium_action_has_multimedia_tracker	multimedia_tracker_tracker_id	On Cascade	On cascade	If multimedia_tracker_tracker_id is deleted then it should be deleted from premium_action_has_multimedia_tracker table
student_plan	premium_premium_id	On Cascade	On cascade	If premium_id is deleted, student

				plan should also be deleted
group_plan	premium_premium_id	On Cascade	On cascade	If premium_id is deleted, group_plan should also be deleted
individual_plan	premium_premium_id	On Cascade	On cascade	If premium_id is deleted, individual_plan should also be deleted
profile	group_plan_group_plan_id	On Cascade	On cascade	If the group_plan_group_plan_id then it should be deleted in the profile table
manifest_has_CDN	manifest_manifest_id	On Cascade	On cascade	If the manifest_manifest_id then it should be deleted in the manifest_has_CDN table.
manifest_has_CDN	CDN_cdn_id	On Cascade	On cascade	If the CDN_cdn_id then it should be deleted in the manifest_has_CDN table.

Section X : Testing

Entity	SQLQuery	Pass/Fail	Error Description	Possible Solution
mailing_list_users	Delete	Fail	Unknown column 'mailing_list_users' in 'where clause'	Typed mailing_list_users after where clause instead of mailing_list_id, change it to mailing_list_id instead.
mailing_list_users	Update	Pass	None	None
generaluser_has_messaging_list	Delete	Pass	None	None
generaluser_has_messaging_list	Update	Pass	None	None
has_logged_from	Delete	Pass	None	None
has_logged_from	Update	Pass	it has passed,However, I realised that IP_validation should only be true/false but I could enter 3.	Change the datatype of Is_ip_validated
registereduser	Delete	Fail	Spelling mistake in column name admin_registered_user	Change it to correct one!
registereduser	Update	Fail	Missed the where clause and changed all the registered user name to the same name	Include complete UPDATE clause
registereduser	Update	Fail	Also realised that registered_use	Change the keys in EER to be unique.

			r_id being primary key still lets me enter non unique numbers as IDs	
admin	Delete	Pass	None	None
admin	Update	Pass	None	None
account	Delete	Fail	region_id got deleted even though it was set to no action	Re-check the relationship in eer and figure out if I missed something while defining
account	Update	Pass	none	none
account_has_role	Delete	Fail	Was set to restrict on delete but the account still got deleted	Change the constraint to no action
account_has_role	Update	Pass	none	none
basic	Delete	Delete	Tried to delete multiple tables together	Follow correct syntax
basic	Update	Pass	none	none
premium	Delete	Pass	none	none
premium	Update	Pass	none	none
advertisements_linked_to	Delete	Pass	none	none
advertisements_linked_to	Update	Fail	Couldn't find the column because of a spelling mistake	Change names to be same as table name
advertisements_has_promotions	Delete	Pass	none	none
advertisements_has_promotions	Update	Pass	none	none

advertisements_has_organization	Delete	Pass	none	none
advertisements_has_organization	Update	Pass	none	none
Bank account	Delete	fail	Incorrect sql query	Column didn't exist because i misspelt.
Bank account	Update	fail	Out of range for column account_number	Change the datatype and set better constraints
paypal	Delete	pass	none	none
paypal	Update	pass	none	none
credit_card	Delete	pass	none	none
credit_card	Update	fail	Out of range value for column 'credit_card_number' at row 1	Change the datatype and set better constraints
debit_card	Delete	Pass	none	none
debit_card	Update	fail	Out of range value for column 'debit_card_number' at row 1	Change the datatype and set better constraints
CDN	Delete	pass	none	none
CDN	Update	pass	none	none
Multimedia_Content_has_CDN	Delete	pass	none	none
Multimedia_Content_has_CDN	Update	pass	none	none
podcast	Delete	pass	none	none
podcast	Update	pass	none	none
music	Delete	pass	none	none

music	Update	fail	Wasn't accepting the year format	Change or restrict to the year format
audio_books	Delete	pass	none	none
audio_books	Update	pass	none	none
published_podcast	Delete	pass	none	none
published_podcast	Update	pass	none	none
published_music	Delete	Fail	Unknown column	Wrongly copied the column name
published_music	Update	Pass	none	none
published_audiobooks	Delete	pass	none	none
published_audiobooks	Update	pass	none	none
Basic_can_stream_music	Delete	Fail	Unknown column	Check the eer to recheck the relationships and names of tables
Basic_can_stream_music	Update	Fail	Unknown column	Check the eer to recheck the relationships and names of tables
Basic_can_stream_podcast	Delete	pass	none	none
Basic_can_stream_podcast	Update	Fail	Got updated in the primary key not keeping the column unique	Change constraints on update to restrict for basic_id
premium_action	Delete	fail	Logically incorrect as it was set on restrict and not on cascade when it got deleted	Change the constraint to set null or no action on delete
premium_action	Update	pass	none	none

student_plan	Delete	pass	none	none
student_plan	Update	pass	none	none
group_plan	Delete	fail	Unknown column	Check the eer to recheck the relationships and names of tables
group_plan	Update	fail	Unknown column	Check the eer to recheck the relationships and names of tables
individual_plan	Delete	pass	none	none
individual_plan	Update	pass	none	none
profile	Delete	pass	none	none
profile	Update	fail	Duplicate entry for role_id	Make sure it is needed for the query to change or keep the columns unique
manifest_has_CDN	Delete			
manifest_has_CDN	Update	Fail	Duplicate entry for key 'manifest_has_cdn.PRIMARY'	Change some inserts to make testing work