

Fashion Model Description System

Jose Rios

Student ID: 921160471

GitHub Username: Colorbomb1

Checkpoint #	Date Submitted
Checkpoint I	9/18/23
Checkpoint II	10/5/23
Checkpoint III	10/25/23

Table of Contents

Project Description	3
Use Cases	4
Functional Database Requirements	7
Non-Functional Database Requirements	10
Entity Relationship Diagram	12
Entity Description	13
EER (Entity Establishment Relationship Diagram)	14
Constraint Description	14

Project Description

The main goal for this Fashion Model Description System is to improve the efficiency and accuracy of recruiting models into the fashion design world. In fashion design, designers need specific models with certain heights and proportions and with this system, it makes sure the designers are choosing the right model for their needs. This will help model recruiters find the individuals they need to best fill out their clothes and look exactly for what they need in a model. This database will take into account different physical characteristics of the model so model recruiters can find models through specific traits they need such as height, weight and the list goes on. On the users end, looking for specific traits in a model becomes much easier because they will be able to filter through all models available, filtering out models that are not in the current scope for that recruiter. As for the models, their unique traits and characteristics will not be overlooked if they are very sought after in the fashion design world. With these models, they also have the option to include what companies and teams they have worked with. These companies also have the ability to create projects that will assign employers to be in charge of and they can invite models to. Not only are cosmetic traits stored but cooperative traits as well, such as where they are located and what languages they speak, making it so that the model recruiter can not only find what they are looking for, but also filter for features that will help them be more cooperative with their models. This system not only increases the search for a perfectly needed model but also finds a much more cooperative one as well. This decreases the time it would take for model recruiters to find the right model and increases the cooperativeness between the fashion designers and the models making the fashion design world one step closer

to a more efficient and better place. A website that would benefit from this system would be www.models.com, incorporating a model recruitment aspect to their website rather than just an informative website about models. Another website that would benefit would be www.lookmodelagency.com, benefiting from this system because they would be able to filter and sort through models depending on the preferences of the modeling recruiter.

Use Cases

1. Use Case: Need model with specific proportions

Actor: Recruiting Model (Jake) - Focusing on Proportional Clothing

Description: Jake is a recruiting model for a fashion design company whose jeans and tops are highly focused on how well their pieces of clothing fits on their clients. They have a fashion show coming up but Jake needs to find people with certain heights and weights to best display their pieces of work. This can only be done by finding models with compatible sizes. Their open modeling auditions weren't cutting it so to fix this issue, Jake needed a place to filter through models with their heights and weights listed, making it much more efficient to find the correct looking models with the right proportions.

2. Use Case: Finding a project whose of interest to you

Actor: Corina - Model

Description: Corina does modeling for a living and although she does not have a tough time looking for fashion shows she would much rather enlist in fashion shows of her interest and which her characteristics are completely compatible

with. Going to open modeling auditions and being invited to work with many companies, it is rare to find a job that she is excited to work with. Corina wishes she can go to a system in which models can request to attend to work for companies rather than her having to wait for requests from companies. With the Fashion Design Description System, requesting and participating in projects will give her a better chance in working with fashion shows she has interest in rather than doing jobs she doesn't want to do.

3. Use Case: Availability within certain region

Actor: Hector - Well Known Model in Mexico

Description: Hector is a well known model located in Mexico who has many companies reaching out to him to request him to participate in their companies fashion design show. The issue is that Hector can only be in one place at a time and most of the requests he is receiving are fashion shows from different countries. Hector would rather only receive requests on fashion shows that are relatively close to his vicinity so that he can stay close to family. To resolve this issue, Hector needs a system such as the Fashion Model Description System which will allow his requests to only be from within his country and filter out all other requests that are not within Mexico.

4. Use Case: Finding models that can be cooperative

Actor: Milo - Italian Speaking Model Recruiter

Description: Milo is a modeling recruiter from Italy, working on a fashion design line that is based on the work of many Italian designers and workers. He is currently located in the United States, expanding the name of their brand to english speaking viewers. Milo's main issue is finding models to help him display his company's clothing due to the strong language barrier between him and the English speaking models in the United States, creating a massive obstacle between the company and its models in this fashion show. To get rid of this compatibility error, Milo needs a system to filter through models that speak the same language, or understand his language to better communicate with his models.

5. Use Case: Making their profile well known

Actor: Julia - Aspiring to become a successful model

Description: Julia is an individual aspiring to become a well known successful model. She believes she has the perfect height and weight to do the job and display clothing from these big name companies. The problem is that everytime she goes to open modeling auditions she spends hours in line and sometimes doesn't even get to be looked at by the recruiters. This causes her to become demotivated to appear at these open modeling auditions even though it's one of her best chances to get her break through. To fix this issue, Julia wants to find a way to have her perfect modeling profile directly in the hands of the modeling recruiters without being behind other models who don't have good characteristics

of her. The Fashion Model Description System will make recruiting her more efficient because other models will be filtered out.

Functional Database Requirements

1. User
 - 1.1. A user shall be able to create at most one account
 - 1.2. A user shall be able to choose multiple roles
 - 1.3. A user must be over 18
2. Account
 - 2.1. An account shall be created by only one user
 - 2.2. An account shall have at least have one role
 - 2.3. An account must have at most one password
 - 2.4. An account must have at most one unique username
 - 2.5. An account must have at least one form of contact
 - 2.6. An account must have one profile picture
3. Role
 - 3.1. A role shall be linked to many users
4. Admin
 - 4.1. An admin is a user
 - 4.2. An admin is the only and only user that can ban accounts
 - 4.3. An admin is the only and only user that can edit other accounts
 - 4.4. An admin is the only and only user that can add/delete roles
5. Model

- 5.1. A model shall be employed by multiple companies
 - 5.2. A model shall be added to multiple projects
 - 5.3. A model must have a height value
 - 5.4. A model must have a weight value
 - 5.5. A model must have a hair color value
 - 5.6. A model shall have a waist size value
 - 5.7. A model shall have a shoe size value
 - 5.8. A model shall have a language preference
 - 5.9. A model shall have a region where they reside
 - 5.10. A model shall have at least one portfolio picture
 - 5.11. A model must have one age
 - 5.12. A model shall have an religion inputted
 - 5.13. A model shall have a biography inputted
 - 5.14. A model shall have a work status
 - 5.15. A model shall have multiple followings
 - 5.16. A model shall have multiple followers
6. Employer
 - 6.1. An employer must be part of at least one company
 - 6.2. An employer must be part of at least one team
 - 6.3. An employer shall create their own company
 - 6.4. An employer shall create their own projects
 - 6.5. An employer shall add other employers to companies
 - 6.6. An employer shall add other employers to projects
 - 6.7. An employee shall request models for jobs
 - 6.8. An employee shall add models to projects
7. Registered Viewer

7.1. A viewer shall have one profile picture

7.2. A viewer shall have multiple followings

7.3. A viewer shall have multiple followers

7.4. A viewer shall have a biography

8. Company

8.1. A company must have at least one employer

8.2. A company must have at least one project

8.3. A company must have at least one model

8.4. A company must have at least one project

8.5. A company shall own another company

8.6. A company shall have one name

8.7. A company shall have at least one location of origin

8.8. A company shall have at least one preferred language

9. Teams

9.1. A team must have at least one employer

9.2. A team shall have at least one project

9.3. A team shall have only one name

9.4. A team shall have employers from multiple companies

9.5. A team shall have at least one model

9.6. A team shall have at least one preferred location of work

9.7. A team shall have at least one preferred language

10. Projects

10.1. A project must be created by a company

10.2. A project must have at least one model

10.3. A project must have at least one employer

10.4. A project must have at most one deadline

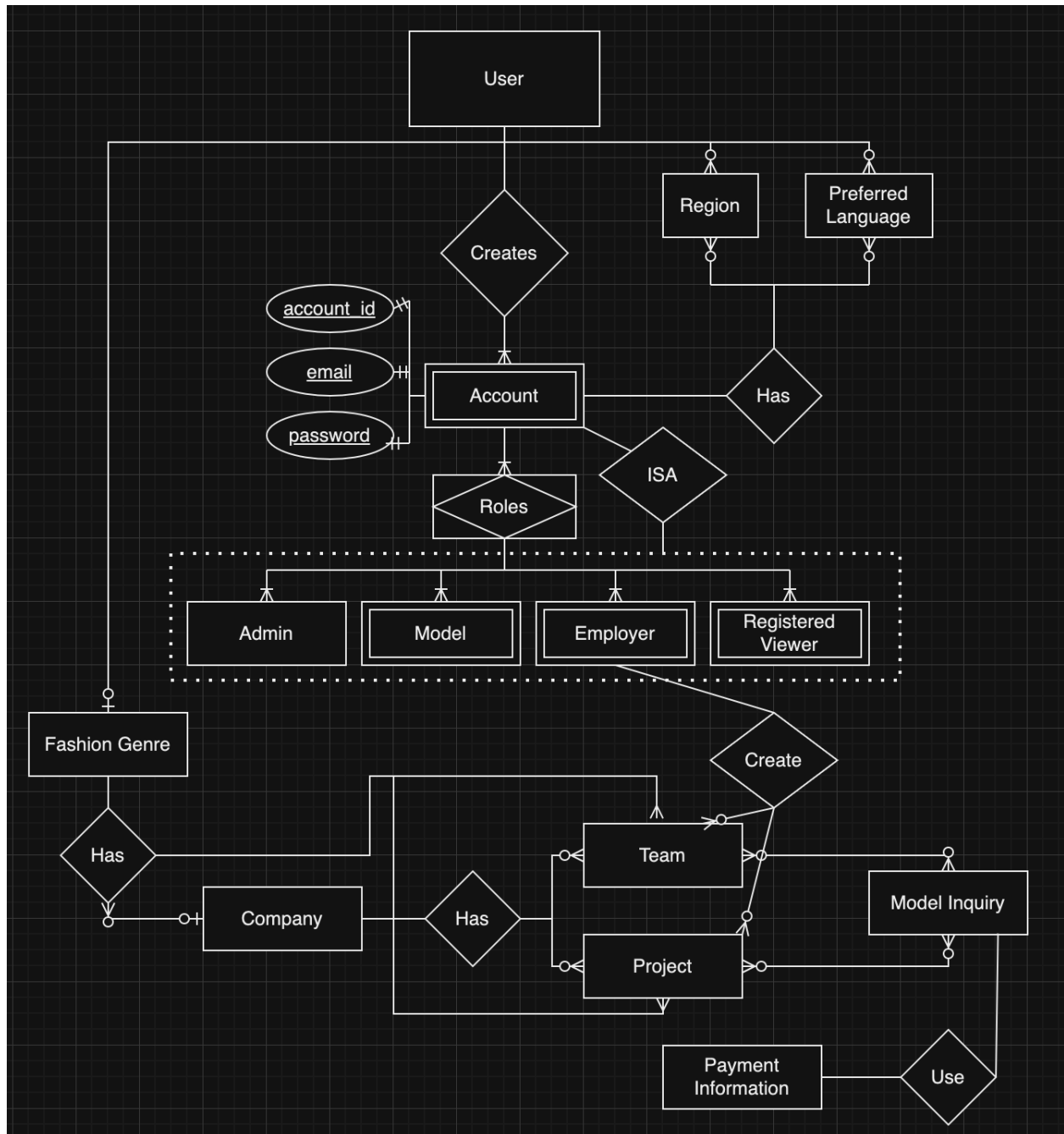
- 10.5. A project must have at least one location
- 10.6. A project must have at least one preferred language
- 10.7. A project must have one name
- 10.8. A project shall have at least one project manager
- 10.9. A project shall have at least one description
- 10.10. A project shall have a preference in models
- 11. Fashion Genre
 - 11.1. A fashion genre shall contain multiple models
 - 11.2. A fashion genre shall contain multiple companies
 - 11.3. A fashion genre shall contain multiple teams
 - 11.4. A fashion genre shall have multiple followers
 - 11.5. A fashion genre shall have a description
- 12. Region
 - 12.1. A region shall have a preferred language
 - 12.2. A region shall have a specified location
- 13. Model Inquiry
 - 13.1. A model inquiry shall have an employer and model id
 - 13.2. A model inquiry shall have a date of when the inquiry was sent
 - 13.3. A model inquiry shall have a expiration date
 - 13.4. A model inquiry shall have a team/project name attached
- 14. Payment Information/Method
 - 14.1. Payment method shall include credit/debit information
 - 14.2. Payment shall include name of card holder
 - 14.3. Payment shall include card expiration date
 - 14.4. Payment shall include billing address

Non-Functional Database Requirements

1. Performance
 - 1.1. The database shall be able to have multiple concurrent recruitments
 - 1.2. Scaling the database larger will have no degradative effects on the database
2. Storage
 - 2.1. The database shall be able to hold all profiles
 - 2.2. The database shall be able to hold all portfolio pictures
 - 2.3. All companies and projects will be stored
3. Security
 - 3.1. All passwords shall be encrypted in the database
 - 3.2. Values stored in the database shall be compatible with the attributes data type and domain
 - 3.3. Database will be backed up every 12 hours
 - 3.4. Account logins will need email verification
 - 3.5. Account contact information will only be accessible if user has an account
4. Usability
 - 4.1. The system will be accessible through different devices
 - 4.2. The system will be accessible through a browser
 - 4.3. System will be available to be used constantly throughout the day expect for when the backup is happening
5. Scalability
 - 5.1. No performance drop will appear when large number of users are using system
 - 5.2. No performance drop will appear when large numbers of users profiles are stored in the system
6. Interoperability

- 6.1. The system will be able to be used through different operating systems due to the system being available on a browser
- 6.2. Being on a computer or a mobile device does not change the accessibility and functionality of the data

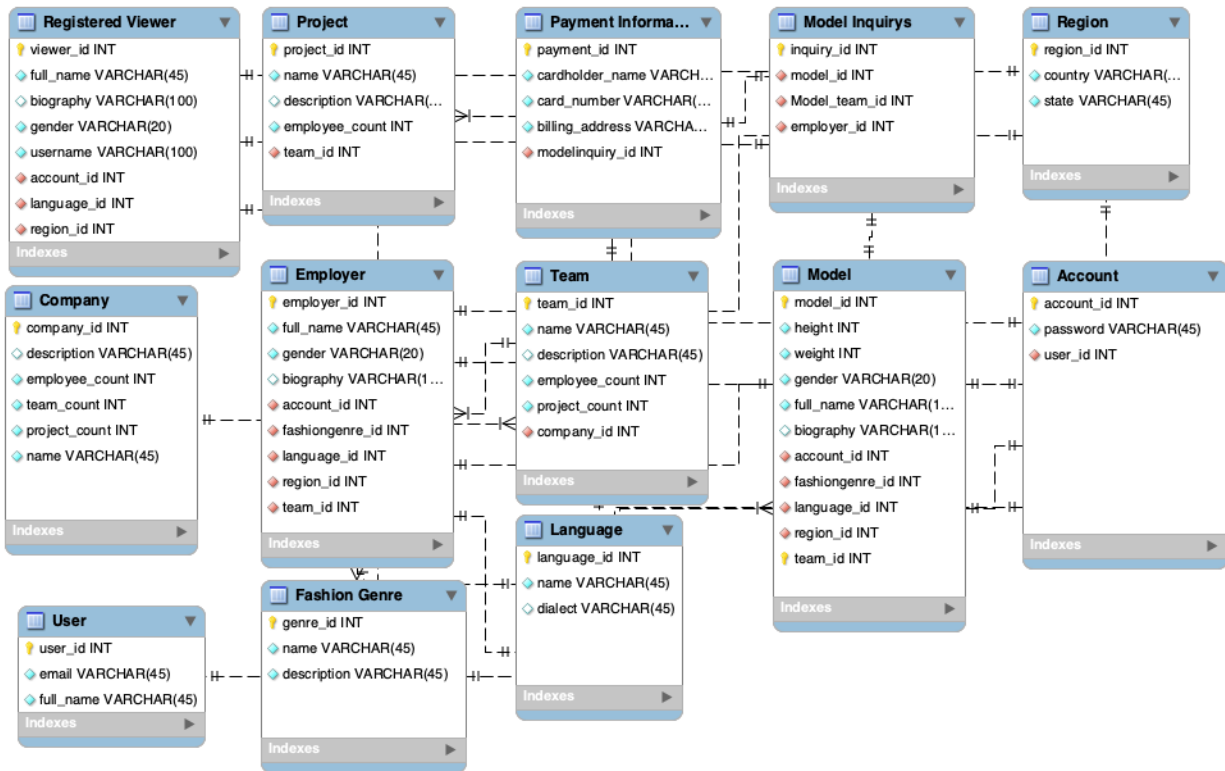
Entity Relationship Diagram



Entity Description

1. User(Strong)
 - a. name: composite, alphanumeric
 - b. email: composite, alphanumeric
2. Account(Weak)
 - a. name_id: composite, alphanumeric
 - b. email: multivalue, composite
 - c. password: multivalue
3. Model(Strong)
 - a. height_id: key, numeric
 - b. weight_id: key, numeric
 - c. haircolor_id: alphanumeric
4. Employer(Strong)
 - a. name: composite, alphanumeric
 - b. email: multivalue, composite
5. Company(Strong)
 - a. company_name: composite, alphanumeric
 - b. date_created: key, numeric, multivalue
6. Project(Strong)
 - a. employ_amount: key, numeric
 - b. date_created: key, numeric, multivalue
 - c. project_name: composite, alphanumeric
7. Team(Strong)
 - a. team_name: composite, alphanumeric
 - b. team_size: key, numeric
8. Region(Strong)
 - a. state_id: composite, alphanumeric
 - b. country_id: composite, alphanumeric
9. Preferred Language(Strong)
 - a. language: composite, alphanumeric
10. Fashion Genre(Strong)
 - a. fashion_name: composite, alphanumeric
 - b. description: alphanumeric

EER (Entity Establishment Relationship Diagram)



Constraint Description

Table	FK	ON DELETE	ON UPDATE	COMMENT
Account	user_id	ON CASCADE	ON CASCADE	If a user is deleted, then the account from that user must be deleted as well