Members of CaiCaoLei:

Jiahao Cai

Bowen Lei

Zhiyuan Cao

Section: Dr Amor

October 12, 2017

Task1 Database Project Proposal

Our group decides to work on database about guitar which includes companies, manufacturers, materials, histories or stories of guitars. We call our project as Guitar Utopia, since we want to create a place for everyone who loves the guitar. We image that there are lots of types of guitars and each guitar is made of different materials (woods) by different companies. Musical companies will own their stores or sell to retailers which we could also keep track of. Companies may also own celebrity endorsements to help sell the products. Some rare guitars with high qualities may even be exhibited or borrowed for show. The reason why we decide to work on this topic is that one group member really likes guitars and he finds it would be very amazing and convenience to have a website where he can get all information about guitars. We believe that this database is also useful for companies to learn the popularity and sale volume of guitars and adjust sale strategies.

Project Title: Guitar Utopia

**Users:**

1. DB Administrator

2. Guitarist

3. Guitar fan

4. Celebrities

5. Musical Instruments Companies

6. Guitar material supplier

7. Guitar Retailer Companies

8. Museum

**Use Cases:**

1) Search for guitar – type

a) User queries the System for different guitar types they like

b) The System return all guitars that matches the specific searching guitar type

2) Search for guitar – material

a) User queries the System for different materials of guitar they want

b) The System return all guitars that used these materials to be produced

3) Search for guitar – appropriate level

a) User queries the System for different levels of guitar to use, like amateur or professional

b) The System return all guitars that fit this level to use

4) Search for guitar - celebrities

a) User queries the System for guitars which played or recommended by celebrities

b) The System return all guitars that are played by certain celebrities

5) Search for guitar - Musical Instruments Companies.

a) User queries the System for guitars which designed by certain instrument companies

b) The System return all guitars that are produced by specified companies

6) User Registration

a) The user places the request to be a member

b) The System notifies the DB Admin that a request has been made from the user to become a member

c) The DB Admin approves the request

d) The System notifies the user

7) Celebrity Registration

a) The user places the request to be a member of celebrity

b) The System notifies the DB Admin that a request has been made from the Celebrity to become a member

c) The DB Admin approves the request and mark the user specially as celebrity

d) The System notifies the celebrity

8) Company and Retailer Registration

a) The companies and retailers who want to join the company list request the registration

b) The System notifies the DB Admin that a request has been made from the company or retailer to become a company member

c) The DB Admin approves the request

d) The System notifies the company or retailer

9) Company and Retailer update information

a) Once company or retailer registration is approved officially, company can add information about company or retailer

b) The System will record these information

10) Company or Retailer deletion

a) Company or retailer decide to delete companies or retailers from the listings

b) The System notifies the DB Admin that a request has been made to delete the company or retailer

c) The DB Admin approves the request

d) The System notifies the company or retailer

11) Cooperation with company and retailer

a) Company can add retailers to cooperation list as recommended retailers

b) System updates the cooperation list of company

12) Retailer joins company’s cooperation list

a) Retailer can send a request to become official store

b) The System notifies the Company Representative, Company Representative can either approves or declines the request

c) The System notifies the Retailer whether his request was approved or not

13) Add guitars to favorite list

a) User can add guitars to its favorite list

b) System updates the favorite and notifies the user.

14) Look for purchase information of guitars for customer

a) User searches his favorite guitar in the system

b) System return all types of guitar users like

c) User chooses one or more guitars to purchase

d) System provides information of sellers that sell these guitars for users

e) User go into these company’s website or connect sellers to place order by information provided

15) Museum posts the information of exhibition of guitar

a) Museum can post the information of exhibition of guitar

b) The System will record the information of exhibition and notifies users interested

16) Give comment of the guitar

a) The users can give comment to a certain data

b) System notifies the DB Admin of a new comment

c) DB Admin either approves or declines the comment

d) System notifies the user whether his comment was approved or not

17) Search for guitar - releasing date

a) User queries the System for guitars by releasing date of guitar

b) The System return all guitars released by that date

18) Posts new product preview

a) Company can posts announcements about upcoming guitars

b) The System notifies the DB Admin that a request of an announcement was posted by company

c) The DB Admin either approves or declines the request

d) System notifies the user whether his request was approved or not

19) Edit guitar information

a) User edit information about guitar

b) The System notifies the DB Admin that an edit request was posted by user

c) The DB Admin checks the edit and either approves or declines the request

d) The System notifies the user whether his edit was approved or not

20) User can buy guitar from Retailer

a) User can ask request of buying guitar from certain company

b) The System notifies the Retailer Representative

c) Retailer representative will check information and place the order

d) The System notifies if the user places order successfully or not and records the

order if it is approved

21) User can borrow guitar (expensive and famous) from company

a) User (Museum or celebrity often) can ask request of borrowing certain guitar

from certain company

b) The System notifies the Company Representative

c) Company representative will check and either approves request or declines it

d) The System notifies if the user borrows the certain guitar successfully or not and records the data if the request is successful

**Database**: Relational database for data storage

**Software**: MySQL Workbench

**Language**: SQL

Are there any machine restrictions for the project?

We haven’t found any machine restrictions right now, but we need to think about how to collect data of guitars and guitar companies. Moreover, if we want to make this database as website, we have to learn how to do that.

**PROJECT** **PROGRESS** **REPORT TASK 2**

Please use the project proposal as your starting point for the progress report. This progress report is an evolution of the project proposal. Please update your project proposal to answer all questions posed in the proposal’s feedback. (1 point)

1. Answer some questions from proposal’s feedback:

**1) Impressive set of use cases and interesting topic (first few functionalities can be ideally combined into single one, ).**

Answer: We tried to combine similar cases into one, so we concluded use cases as following: users registration, search guitars, update guitar, company and retailer store data, company or retailer store data deletion, user updates data about themselves, company representative posts sale information cooperated with retailer stores, posts new products preview.

We decided to drop use cases of No.20 and No.21, since our database only records the general information of guitars, not concrete guitars.

**2) Based on point number 1, this would be a website as you have specified. Identify your infrastructure quickly because this implementation would take time.**

We identified basic and brief infrastructure of guitar database and drew the UML diagram firstly. After that, we implemented the graph to relational model in MySQL.

**3) Would you plan for role base navigation (i mean in case retailer is logged in the functionalities would be different and for a normal user view of website would be different)? Also, would there be any restriction on data access for each user?**

Answer: We designed our database as an information platform of guitars. We will have four different users which are normal users, and representatives of instrument companies, retail stores or museums. Normal users can access to all data of guitars and they could leave their comments of guitars. Normal users could also update data of guitars and these updates will be checked and approved by DB Administers. They could also search all information of instrument companies, celebrities, retailer stores and museums where will exhibit guitars. They could also check all data from sale Info, endorsement, produce Info and exhibition, but they are prevented from editing those data. Only companies’ representatives and retailer stores’ representatives can update data o their companies or stores approved by DB Administers. By using this database, normal users can easily search certain guitars they want with all information, like prices in different stores, released date, designer’s name and so on. They could then access to websites of different companies or stores to buy or check more details. They will also receive the information of exhibition of guitars they like to get the latest news of guitar in their favorite lists.

Furthermore, we provided different views for different users. Users can only see and edit their own private information, but anyone can access to basic information about others.

**4) You should identify some basic set of relations for your schema because again you have an impressive list of use cases and it would require a precise implementation.**

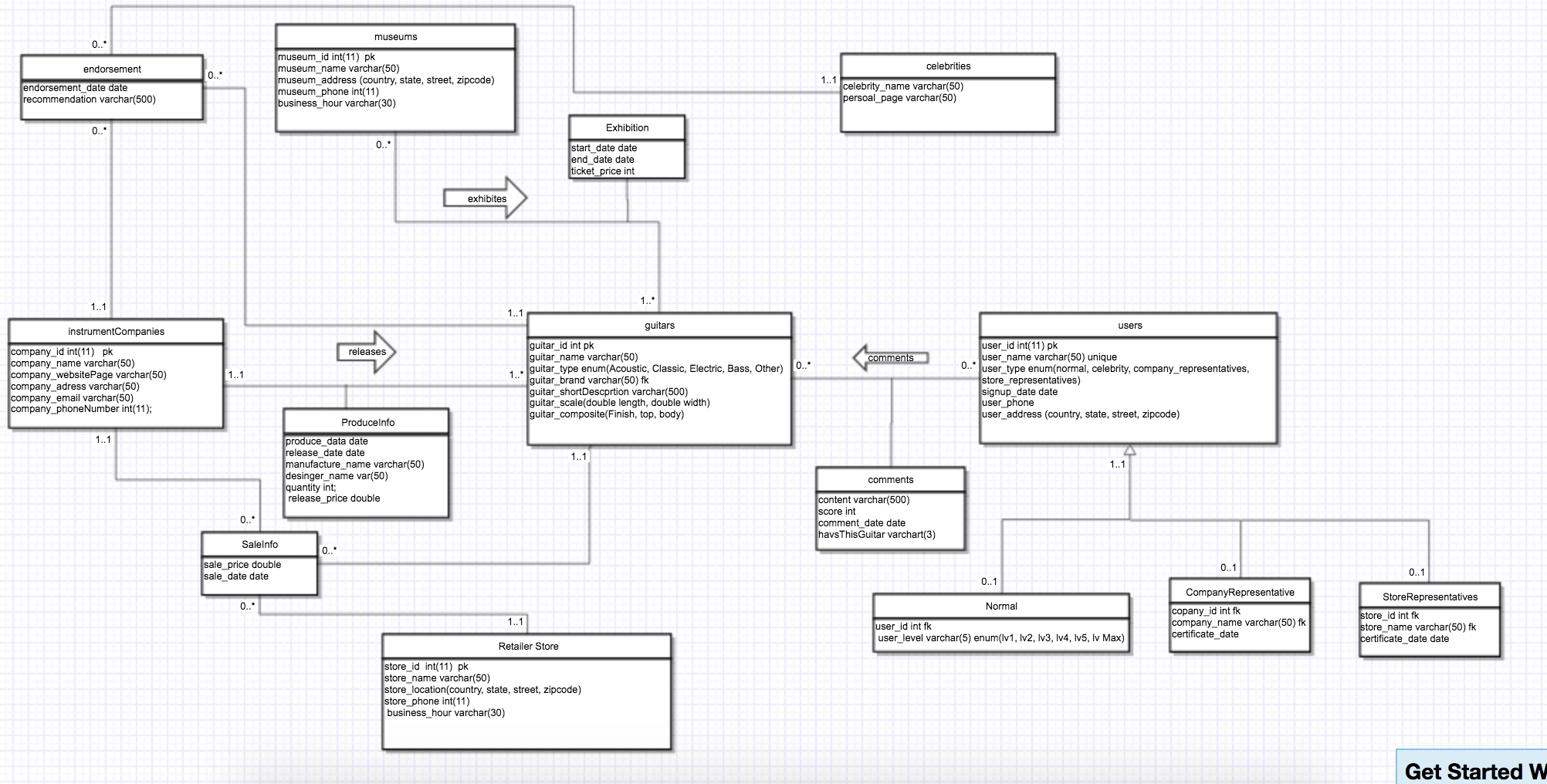
We have identified basic information and attributes about guitars, users, companies and so on. All of them will be used for searching or editing based on list of use cases.

**5) How the database is managed, inclusion of new products, exclusion of old/damaged products etc.**

Our database only recorded general information for different types of guitar instead of data of every specific guitar. The broken or damaged guitar products information will not be managed in our database.

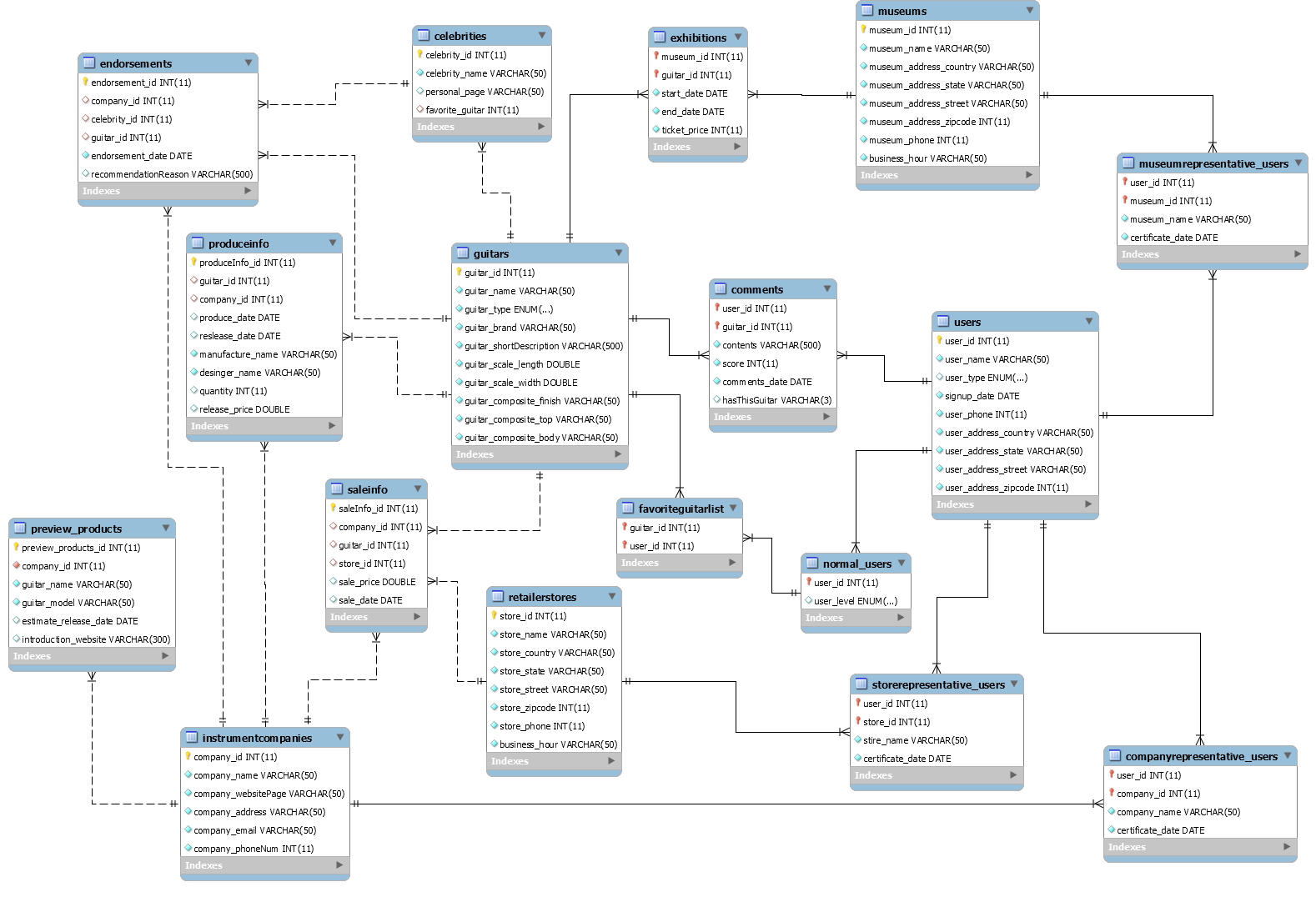
2.An UML diagram of the database which you will be using for the project.

It’s only a draft. When we implement our database schema, we add more tables and foreign keys. Please check our EER diagram in problem 3 with more specific entities, attributes, relations, primary key and foreign key.



3. An EER diagram converting the UML diagram to a relational schema.

We used mysql to create database and table schema and convert it to this EER with all entities, attributes, relations, primary key and foreign keys.



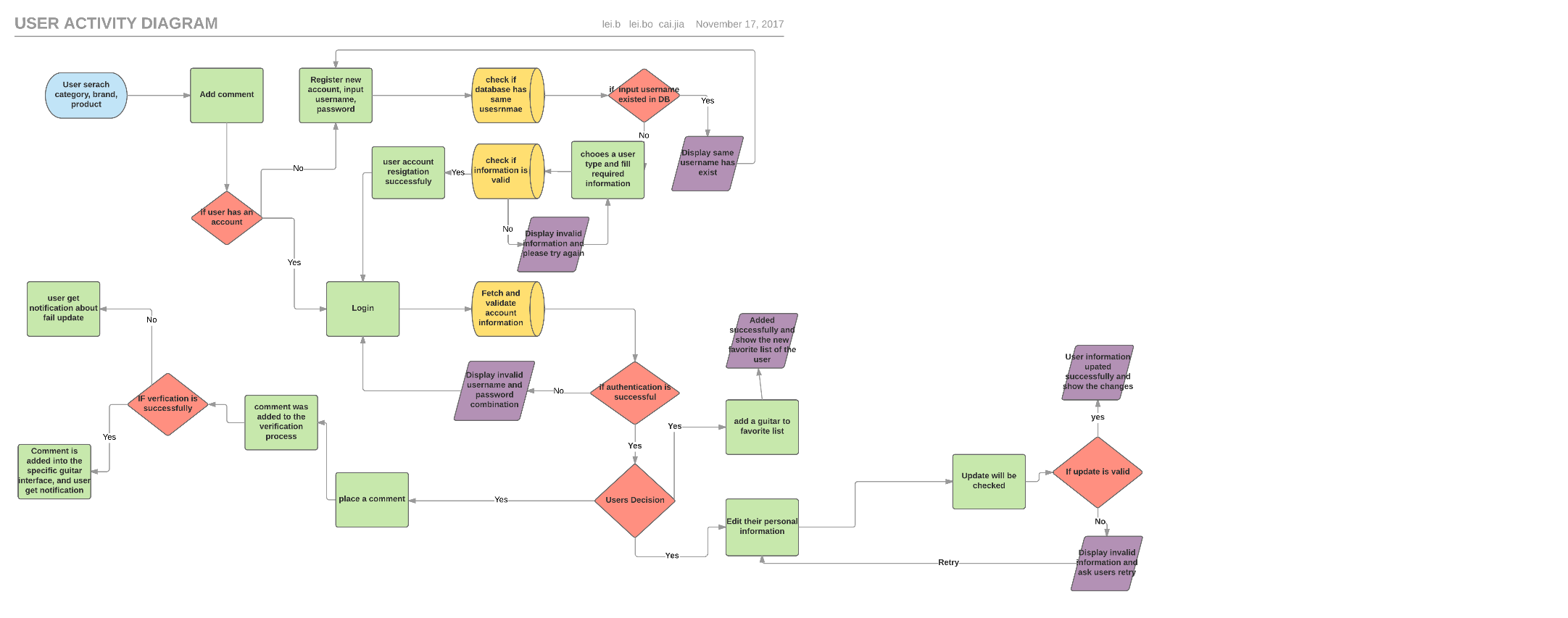
4. A brief step by step user interaction of your application, including all CRUD operations (create, read, update, delete) you plan on implementing for the project.

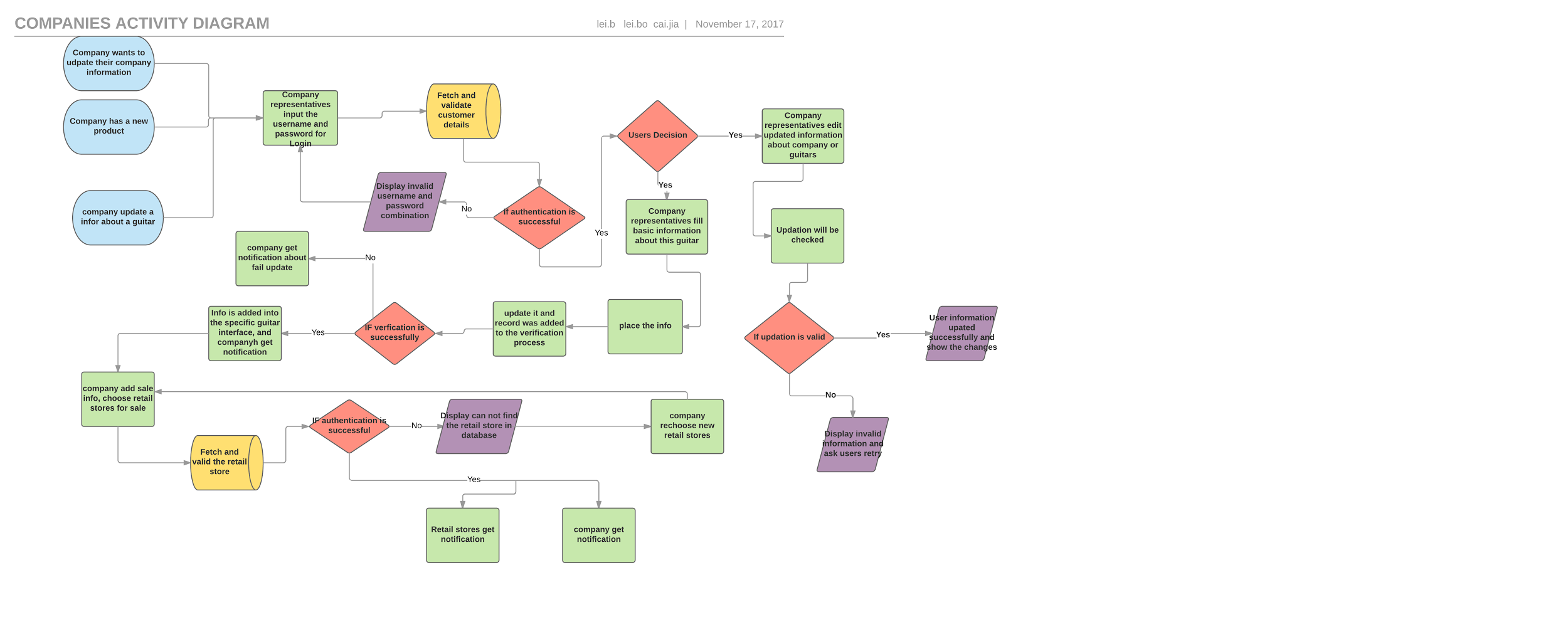
To implement user interactions, we separate them into three parts including user’s interaction, company’s interaction and museum’s interaction.

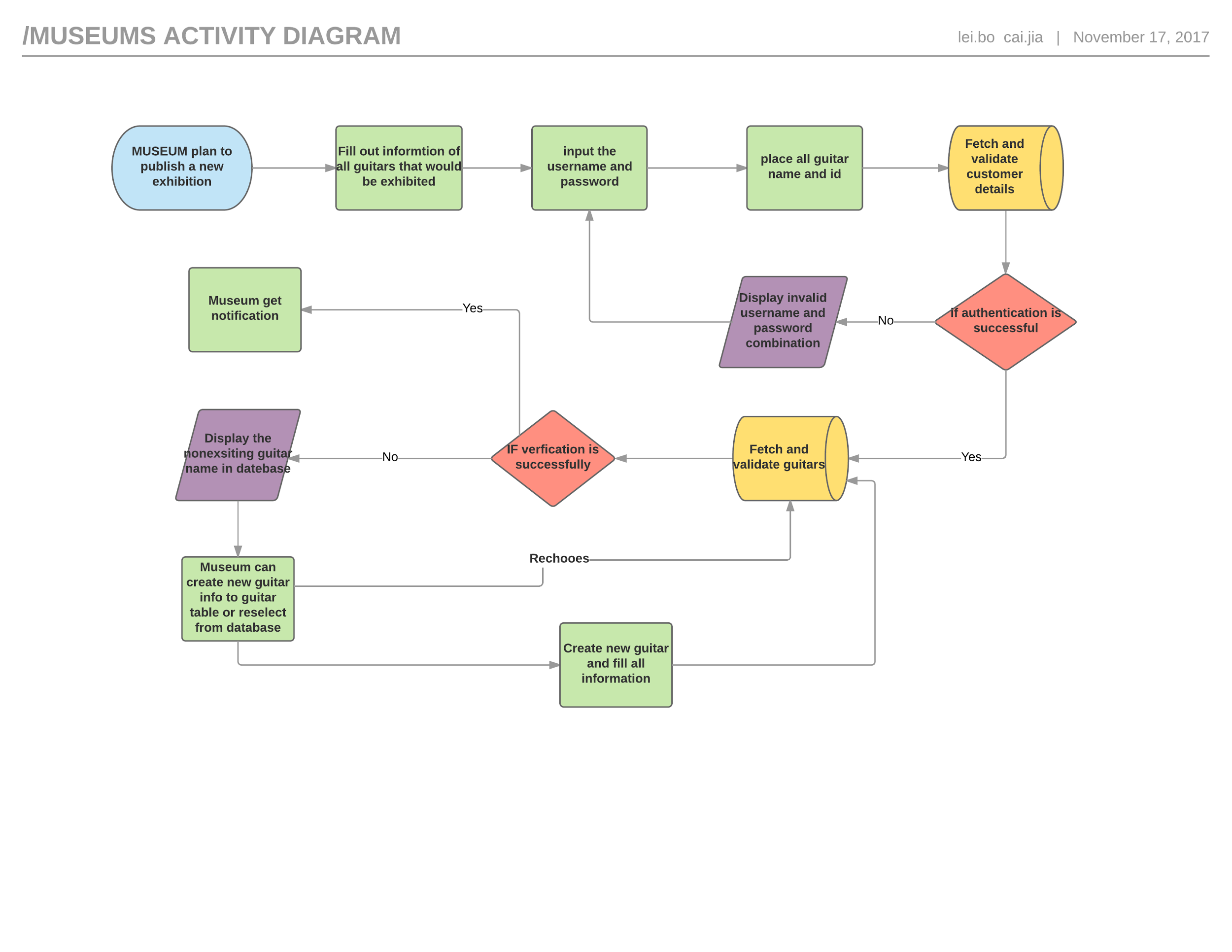
The normal user will register their accounts, the DB will check its user type and approves the registration. Any user could search data of guitars, instrument companies and store. After users have their accounts, they are able to edit their portfolio so they will upload information to database and will be approved by DB or not. Users can add guitars to their favorite list.

Company will ask representative to post their new products and update company information, and they also have authority to update guitar information.

Museum can ask their representative to post exhibition of guitars and they are allowed to add their old guitars from their archive.







5.

For database we will use SQL and MySQL workbench 6.3.

If we do want to create a website, we will try to use:

Build tool: .NET Framework in Visual Studio 2017

Web Develop Language: Front-end language: HTML5 for contents of page and CSS version 3.0 for website style, and back-end language: and C# 3.0 for functionality.

**Group CaiCaoLei Project Final Report Task 4**

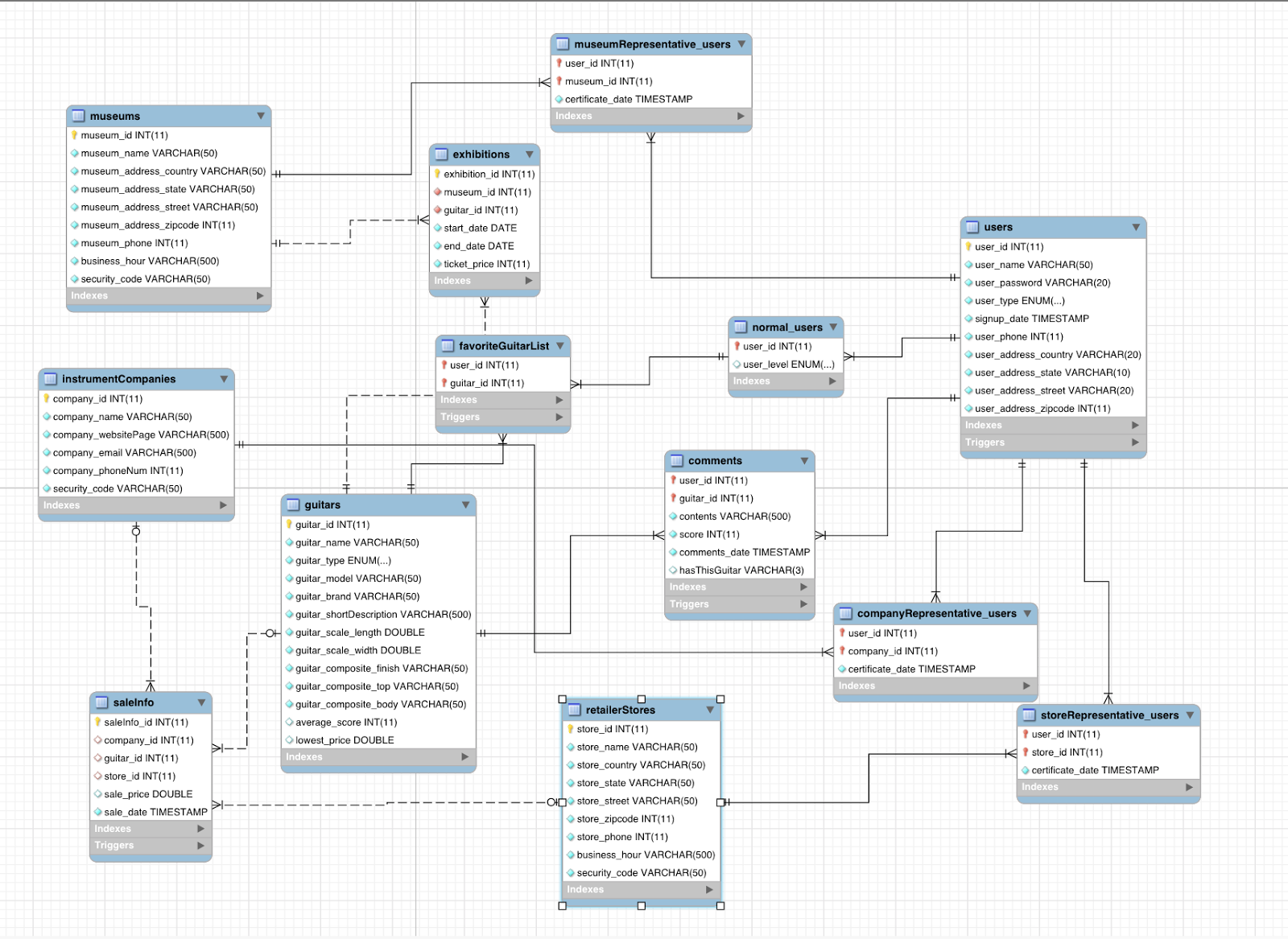
**1.   Provide a README section for creating and running the project. Provide complete specifications for building your project. Specify all libraries, software, etc. needed to run the application. Specify expected installation directories. If you use a specific technology for the project, the technology’s download page must be listed.**

Our back-end part is mysql database schema and all data is in one submitted sql file. Our front-end part is using php and wamp to run in localhost. Firstly, because wamp only works for windows computer, if your computer is max, please use the xampp. The following instruction is in windows. Firstly, you should download the wamp: <http://edu.metinfo.cn/upload/file/wamp-64b.zip> and if your computer has mysql, please change the port of wamp to another port not 3306, please follow directions: <https://stackoverflow.com/questions/37262216/mysql-server-5-7-and-wampserver-in-the-same-machine-wamp-cant-run> And you can extract our zip “Guitar\_Website\_Front\_End” into your www file that is  in your wamp file. Later, you should open our “Guitar\_Project.sql” file in your MySQL and run it to create database and  you should open “Connect\_Database.php” to change the password and user-name of your database connection. Next, you can open browser on this url: <http://localhost/Guitar_Website/> , click [DIR] Documents, and click Start\_Guitar\_Database to start the guitart website.

**2.   Provide the Technical Specifications (as defined in the progress report) for the project.**

For database we will use SQL and MySQL workbench 6.3.

For frontend website, we use php and HTML as front end language and use WAMP as local server.

**3.   Provide the final UML/EER for the project or the submitted database schema (Reverse Engineer your final schema in the MySQL workbench). Include any comments you have in order to explain your design choices regarding the schema, tables or the attributes within.**

One difference from our initial design is that we delete tables of celebrity and endorsements because we think our data schema is huge and we cannot add some useful triggers for these two tables. Our data schema still is huge. We have users table with three subclasses to reduce the duplicated code. They are normal users, company representatives, store representatives and museum representatives. We have our sale information of guitars which involves in guitars table, companies table and stores table, because we think a guitar will be released by a company and be selled by a store.

Another change for our database schema is we add a attribute called security code in companies table, stores table and museums table, because we think representative users  should be issued by his company or store or museum. When these types of users want to sign up a representative account, we need to check these security code.

We also add attributes of lowest price and average score in our guitars table, since we also want to update these two attributes when new sale information are posted or new comment is posted. (We love triggers.)

**4.   Provide the final user flow of the system. List the commands or method the users performs to interact with the system.**

1. Register: user should chooes one of types to register their account.
2. Log in: user use their account to log in the system, system will check the validity of the username and password.
3. Add guitar(only company and museum), they can give all information about data.

           In our system, there are some restrictions to verify guitar available, for instance,    length of a guitar can not smaller than 0.

1. Search guitar information: user input id or name to find the characters of a guitar.
2. Edit user information: user has authority to edit their information, including: name, address, phone, email. However, he can not change his id.
3. Show user information: user can get his current all statement from our websites.
4. Comment a guitar: After user search a guitar, he can leave his thinking about the guitar.
5. Add an Exhibition (Only museum): museum can add a exhibition in our website, at the same time, it should pick selected guitars in this exhibition. Moreover, it can add some new guitar information to complete the advertisement information.
6. Add saleInfo: Company or Store can add sale information about a specific guitar for customer, although our website can not offer buying online, but it is free to contact seller directly.

**5.   Provide a “Lessons Learned” section.  For example:**

**1.   Technical expertise gained**

**2.   Group work insights, time management insights, data domain insights etc.**

**3.   Realized or contemplated alternative design / approaches to the project**

**4.   Document any code not working in this section**

1. We have a lot mySQL code which includes DDL and DML, so we think we practice a lot and are expert on that. We create 1 function, 6 triggers, 28 procedures and 1 event. We have only one function, since we don’t really need to write function. For some procedure, we also use cursor, so we think we basically implement all things we learnt on this course. Bowen Lei learnt php and HTML by himself and created the website of our project. He taught us so it was amazing.
2. We didn’t really have any idea at the beginning and we talked a lot together and even argued with each other, but finally we come up with the idea of creating the database of guitar. We learnt that sometimes arguing is bad, but sometimes it will promote our works. We argued since we all passionate about this project and we think we come up with a really good project. We meet up about 2 times a week and it seems like too less, but we all have different time schedule. It’s really challenging for us to manage our schedule. However, everyone would like to meet up as long as they are free. Moreover, we only think about the topic we are interested in instead of thinking about where can we find some existing data collection. Therefore, we have to type data by oursleves. We need to think about this issue in future works.
3. We delete tables of celebrities and endorsements from our initial design, since we don’t realize how huge our data schema is when we only design it. We find problem that it’s too huge to manipulate and it’s so useless to have such design, since we cannot add any triggers instead of only some simple query. We also add some attributes when we create our procedures and triggers. For example, we found we need the security code in our companies table , stores table and museums table to check if the user is issued by company or store or museum. Other than that we think our design and aprroch are good.
4. When we used the cursor, we found that we didn’t put the statements in while loop and the procedure only worked once. Fortunately, we debugged it and learnt how to use cursor and we would always remember it, since we almost spent one day to debug.

**6.   Provide a “Future work” section**

1.Try to insert more data, since we are really interested in that.

2. We will learnt php and HTML.

3. Finish our website server. We will make our user interface better and user friendly. We will implement functionalities we haven’t finished.

4. Figure out how to use other server instead of local server.

5. Perhaps we could add trading system for our database, including selling and buying guitars or selling guitar exhibition tickets.

**7. Briefly describe the contribution of each group member to the design and development of your project.**

1. Jiahao Cai: create the blueprint for this guitar assignment, finish the mysql code compliation.
2. Bowen Lei: charge the consturction about front-end part, connecting database with website for user interaction.
3. Zhiyuan Cao: complete the user case planning, participation connect database and browser part.

**Please include a list of all the procedures, triggers, functions, transactions, error handling  you have implemented as part of the project report, with a brief & concise explanation of their role/purpose. One line explanations are sufficient.**

Please check our README file.

1. logCheck: A function to check the account name and password when users log in. It will return true when user name and password are correct.
2. validUser:  trigger to check if the information users provide is valid when they sign up an account.
3. signUpNormal: A procedure to sign up an account users need to type user name, password, phone number, country, state, street and zip code.
4. insertANormal: A trigger when normal user is signed up, it will help to add this user to normal user table.
5. signUpCompanyR: A procedure to sign up a company representative.  It will ask user name, password, phone number, country, state, street, zipcode, company name and security code issued by that company.
6. signUpStoreR: A procedure to sign up a store representative. It's similar to sign up a company representative, but will ask store name and the security code.
7. signUpMuseumR: A procedure to to sign up a museum representative.  It's similar to sign up a company representative but will ask museum name and the security code.
8. showOwnUserInfor: Users can use this procedure to check its own information according to their user types.
9. editCommonUserInformation: Users can use this procedure to update their personal information. They can edit any field they want. They could leave some field as blank and those blank field will be set as null and won't update the field in the database.
10. showCompanyInforByrepresentative: Given one company name existing in database user want to search , call this procedure to see company information
11. showRetailInforByrepresentative : given one company name existing in database user want to search , call this procedure to see company information.
12. showMuseumInforByrepresentative: given one company name existing in database user want to search , call this procedure to see company information.
13. editCompanyInfor: provide necessary security of company and any information you want to edit and others just remain null.
14. Edit Store Information: provide necessary security of store and any information you want to edit and others just remain null.
15. Edit Museum Information: provide necessary security of museum and any information you want to edit and others just remain null.
16. Leave comments:  After users log in successfully, they can leave comment for guitars. They could only leave one comment and score for each guitar.
17. Trigger: scoreInsert:  updater average score when the one comment is updated for one guitar the range of score should become 0 to 100/
18. Trigger scoreUpdate:  update score of guitar in comments, calculate the average score of guitar again if there is one comment for the same guitar by the same user
19. Add favourite list:-- To add a guitar to favourite list of normal users.
20. Trigger: favouritelistCheck: trigger for preventing users from repeating adding same guitars to their favorite lists.
21. Search Comment: To search comments for given guitar's name.
22. Post Sale: Post the sale info of guitars. It will need company name, store name, and price of guitars.
23. Trigger: updateLowest -- A trigger to update the lowest price of the guitar if there's a new sale posted with lower price.
24. Post Exhibition information: -- This is for museums to post an exhibition of guitars. It will need gutarName, museumName, start date, end date and ticket price.
25. Expired Exhibition : A procedure to check the end date of exhibition drop the expired exhibition.
26. After use postExhibition procedure to post one information that end date is passed to current time, call expiredExhibition, it will delete it automatically
27. Update Level:  A procedure to check the date and update normal users level according to their singup date.drop procedure if exists updateLevel;
28. Guitar\_engine: A event call guitar\_name\_given, guitar\_name\_given every day.
29. Please use set global event\_scheduler = on;  We have tested updateLevel and expiredExhibition successfully, so we think this event also works.
30. createAGuitar: A procedure to insert a guitar information.
31. createAComp: A procedure to insert a piece of company information.
32. createAStore: A procedure to insert a piece of retail store information.
33. createAMu: A procedure to insert a piece of museum information.
34. deleteComments: A procedure to delete the certain comment. Users will specify the guitar name of the comment they want to delete.
35. deleteUser: A procedure to delete the user account. Account could be all types of users.