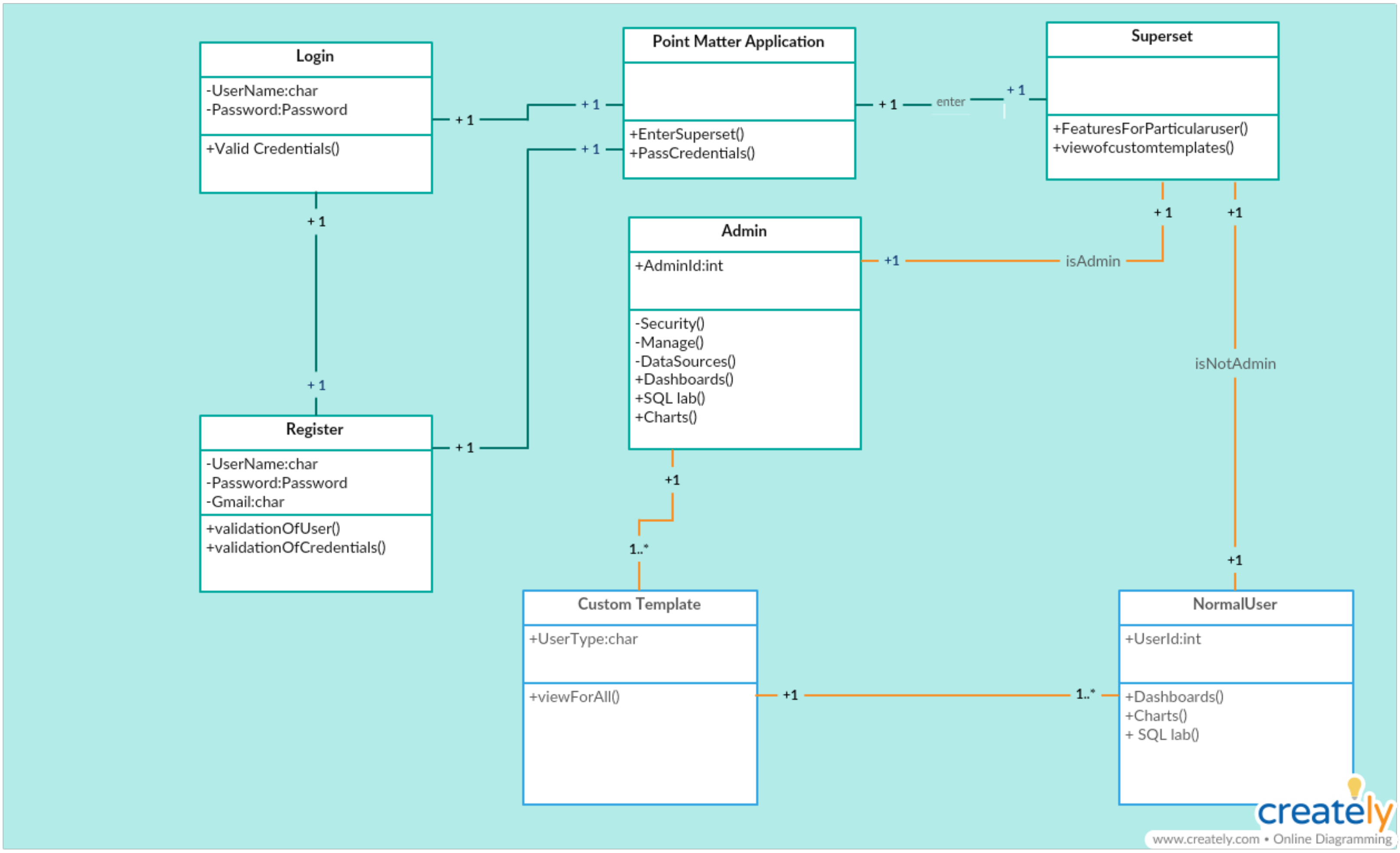
Product Design

Team 12

R.S.Subbulakshmi
Debojit Das
Ujwal Narayan N,
Arnav Kapoor,
Chinni Venkata Charan

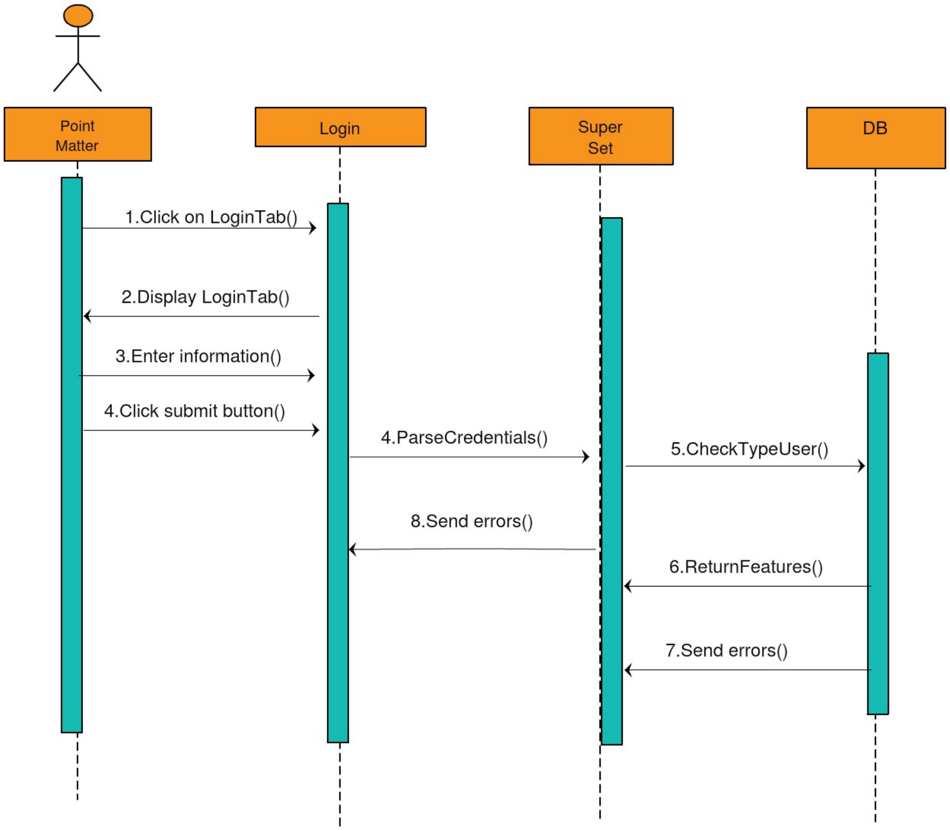
Design Model

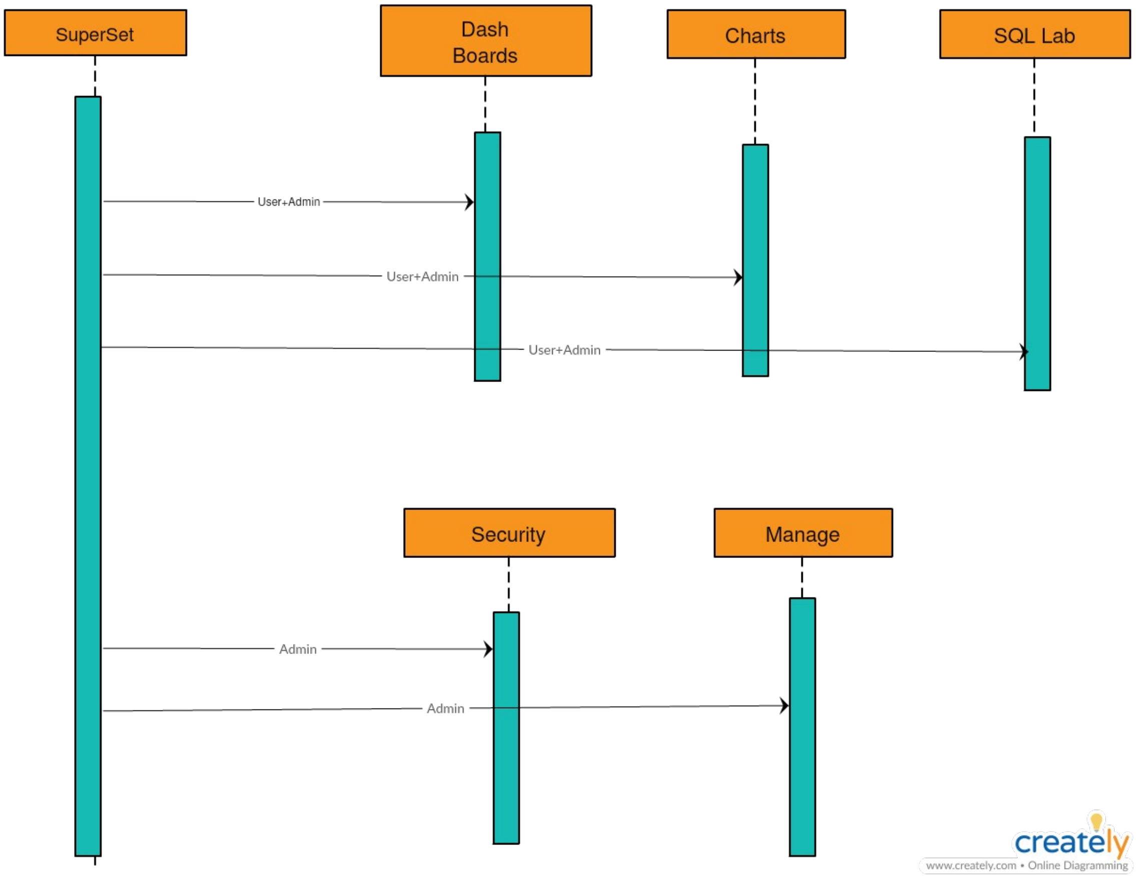
The fundamental aspect of this project is an end user can visualize the data by running the sql queries after login into the application .



Single Login	Login Behavior • Input:credentials of a particular user • Flow:When a particular user login user into the application then he don’t need to again login into into the superset application as we parses the credentials to the superset application • Parses the credentials to the application
Roles Assignment	Creating the roles • Different Persons will have different roles in the applications • Eg: Admin can have additional roles other than normal user like he can view the list of users who logged into the application
Custom Templates	SQL Lab • Through Sql lab which is a part of superset application he can write his own queries and visualize the data based on his requirements Custom Templates • There are a set of custom Templates where we can visualize the data without need of querying the data • We create the templates using the SQL queries based on the requirements
Contribution to Superset	Contribution • This is a application which was built using D3 graphics,react js and also flask-app-builder which have very poor documentation,the team has to go through the code on how it was written and finds the error (if any) • So,now we contribute the fixes to the community which would help them in future

Sequence Diagrams





The above diagrams is a sequence diagram, which explains the flow of control from one part to other.

Design Rationale:

For single login :

Design1 : Use iframes for single login into Superset application from Point matter

- **Advantages :** Seemed easy , iframes well known
- **Disadvantages :** Superset does not allow cross-side scripting .Also passing of credentials makes the login vulnerable . Due to this design we will have to submit 2 applications instead of 1 . Integration will thus become hard .

Design2 : Login by using database – authentication

- **Advantages :** Easy to implement
- **Disadvantages :** Parameter are of different types between the Point matter and Superset application and thus the checking credentials is harder . Also passing credentials makes the application vulnerable .

Design3 : Using OAuth

- **Advantages :** There is no manual checking . Services already provided by the OAuth provider . Google is chosen for the implementation as it is widely known .The home tokens are stored as cookies in the browser .It is not vulnerable as the above options because it is stored in the locally .
- **Disadvantages :** Have to make changes in the source code .

For custom roles :

Design1 : Admin and other users have access to all databases and data-sources .

- **Advantages :** No user has any problem with permissions .Sql lab editor can be used smoothly .
- **Disadvantages : No privacy is maintained . Changes can also be made for th permissions of other users and the main databases**

Design2 : Use custom roles

- **Advantages :** By using the GUI to combine different roles , Gamma user and other specific permissions , the user will have access to only allowed databases and data sources .
- **Disadvantage :** Implementation was hard due to very less documentation and already raised issues regarding the implementation which were either closed or inactive for a long time .

For templates :

Design1 : Manually convert T-SQL to MySQL and create dashboards using Superset application

- **Advantages :** Using GUI
- **Disadvantages :** Manual conversion of T-SQL to MySQL is difficult due to different syntax and Superset application when run on the complete database causes a lot of hang / takes a lot of time .