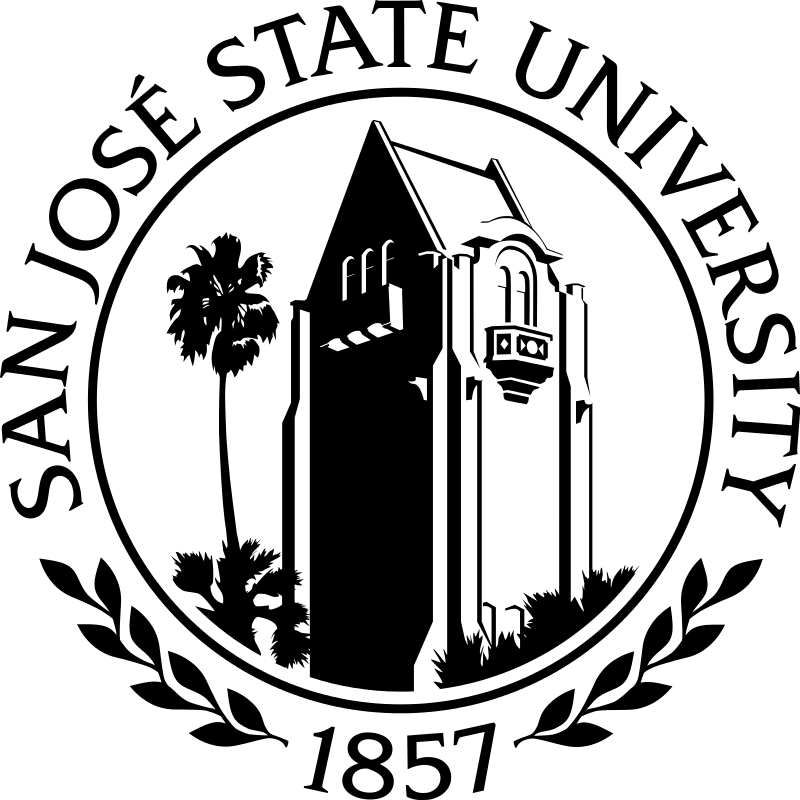
**CS 157A Project Data Model & DB Design Revision**

**Three-Tier Web Application**

**SJSU Bookie**

****

Cole McKinnon, Jonathan Van, Yu Xiu

Team 4

Advisor: Dr. Mike Wu

Oct. 15, 2019

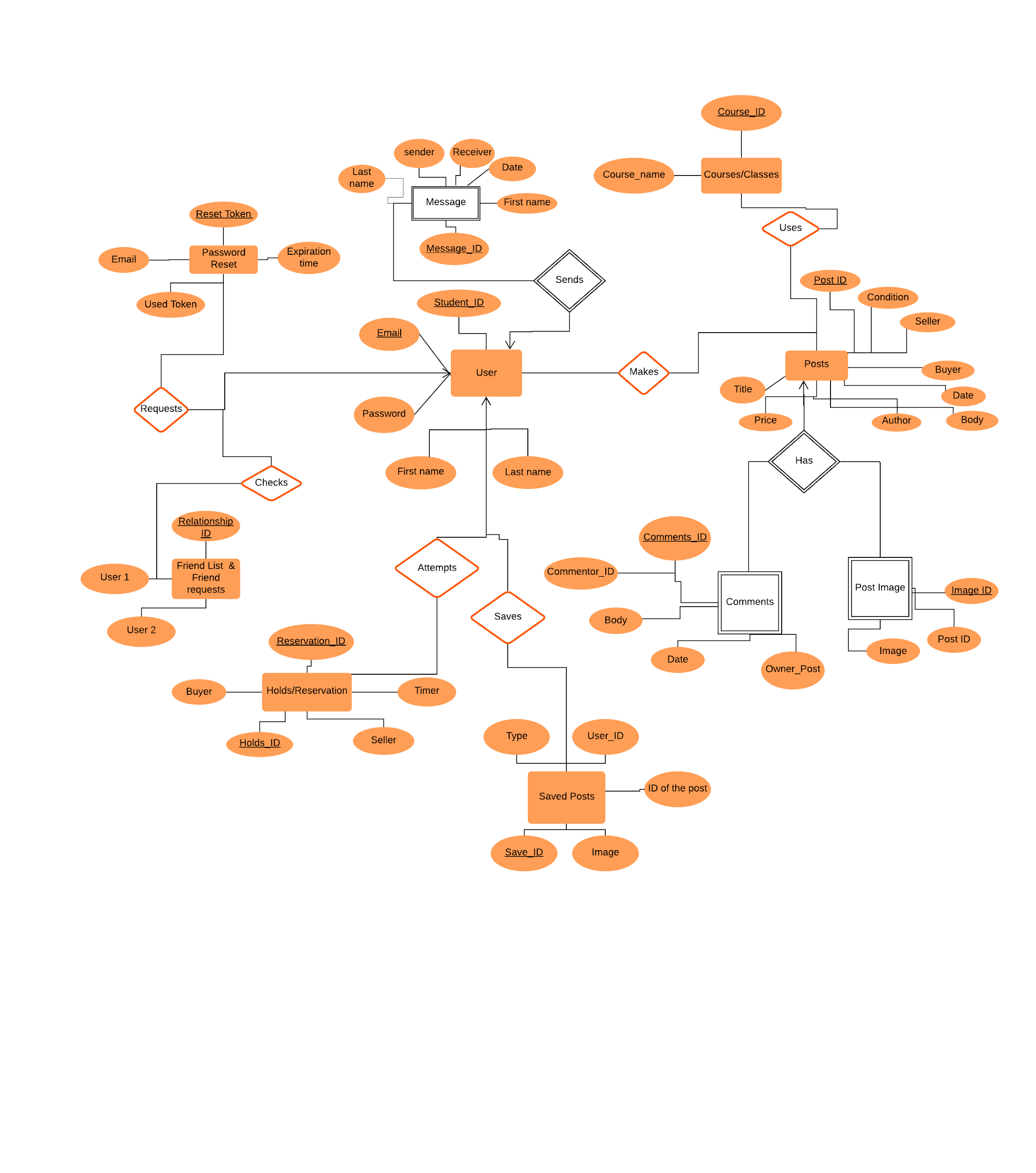


Figure 1: SJSUBookie E/R Revised Diagram

**Explanations for E/R Diagram:**

* **Tables**
  + User
    - The users who are using our SJSU Bookie. The Unique keys are the user’s email and student ID.
  + Password Reset
    - Unique key is the reset token where this allows the user to be able to reset their password in case they forgot it. Token is used as validation and identification.
  + Post
    - Posted by a user, this entity has details of the book they are trying to sell. Another user can put a hold, view or comment on it, but only the original poster can edit/delete the posting. These are recognized by the post id.
  + Saved Posts
    - This is a log/list of posts that is relevant to a user. These posts can include posts that you’ve made, posts that you’ve commented on, or posts that you favorited. These are identified by the post\_id and saved\_id
  + Messages
    - Users can send messages to other users, but the constraint is the person must be a user, then he or she can send messages to other users. The message ID number is the unique key.
  + FriendList/Friend Friend Requests
    - A user is able to add another user as their friend on SJSUBookie in order to maintain communication or be able to communicate with one another. A friend request is considered accepted when both users recognize that the other is the friend on the database. These are uniquely identified by the relationship ID.
  + Holds/Reservations
    - A hold is basically a contract stating that one user is interested in another user’s post. This is temporary let other users know that the book might be bought. Identified by Hold/Reservation\_ID.
  + Comments
    - Users can leave comments under a post. The unique key of comments is comment ID, which we use to identify the comments.
  + Post Image
    - It will be a table of images the user uploaded. The users would upload images when they are creating a selling post. The unique key is the image ID.
* ENTITY RELATIONSHIP EXPLANATION
  + **FORMAT: ENTITY\_1 RELATIONSHIP ENTITY\_2**
  + User requests Password Reset
    - A user will have the option to reset their password if they’ve forgotten it.
  + User Checks User Friend request
    - A user has the option to see all pending friend requests and accept friend requests from other users.
  + User Attempts to buy holds/reservation
    - A user must be able to put a reservation on hold if they are interested in the listing but unable to buy immediately. This will prevent another user from buying the listed book.
  + User Logs Saved Posts
    - Any post the user has saved, posted, or removed will be logged, providing the option to view at a later time.
  + User Makes Posts
    - A user needs to be able to make a post to sell a textbook.
  + Posts have comments
    - Every available post can have comments regarding the book or the seller.
  + Posts have post image
    - A post should have a corresponding image matching the textbook cover.
  + Posts use course/classes
    - Each book posting can have relevant classes associated with to improve search results.
  + User sends messages
    - Buyers and sellers can communicate by sending messages to increase transaction security.

**Schemas:**

* Users(Stdent\_ID, Email, User, Password, First name, last name)
* Message(Message\_ID, First name, Date, Receiver, sender, Last name)
* Holds/Reservation(Reservation\_ID, Holds\_ID, Buyer, TImer, Seller)
* Saved Posts (Save\_ID, User\_ID, post type, ID of the post, imageID)
* Friend List & Friend requests(Realtionship\_ID, User1, User2)
* Password Reset(Reset Token, Email, Expiration time, Used token)
* Posts(Post\_ID, Book\_condition, Seller, Buyer, Book\_Date, Body, Author, Price, TItle)
* Comments(Comments\_ID, Commentor\_ID, Body, Date, Owner\_Post)
* Courses/classes(Course\_ID, Course\_name )
  + - Post Image(Image\_ID, Post ID, Image)