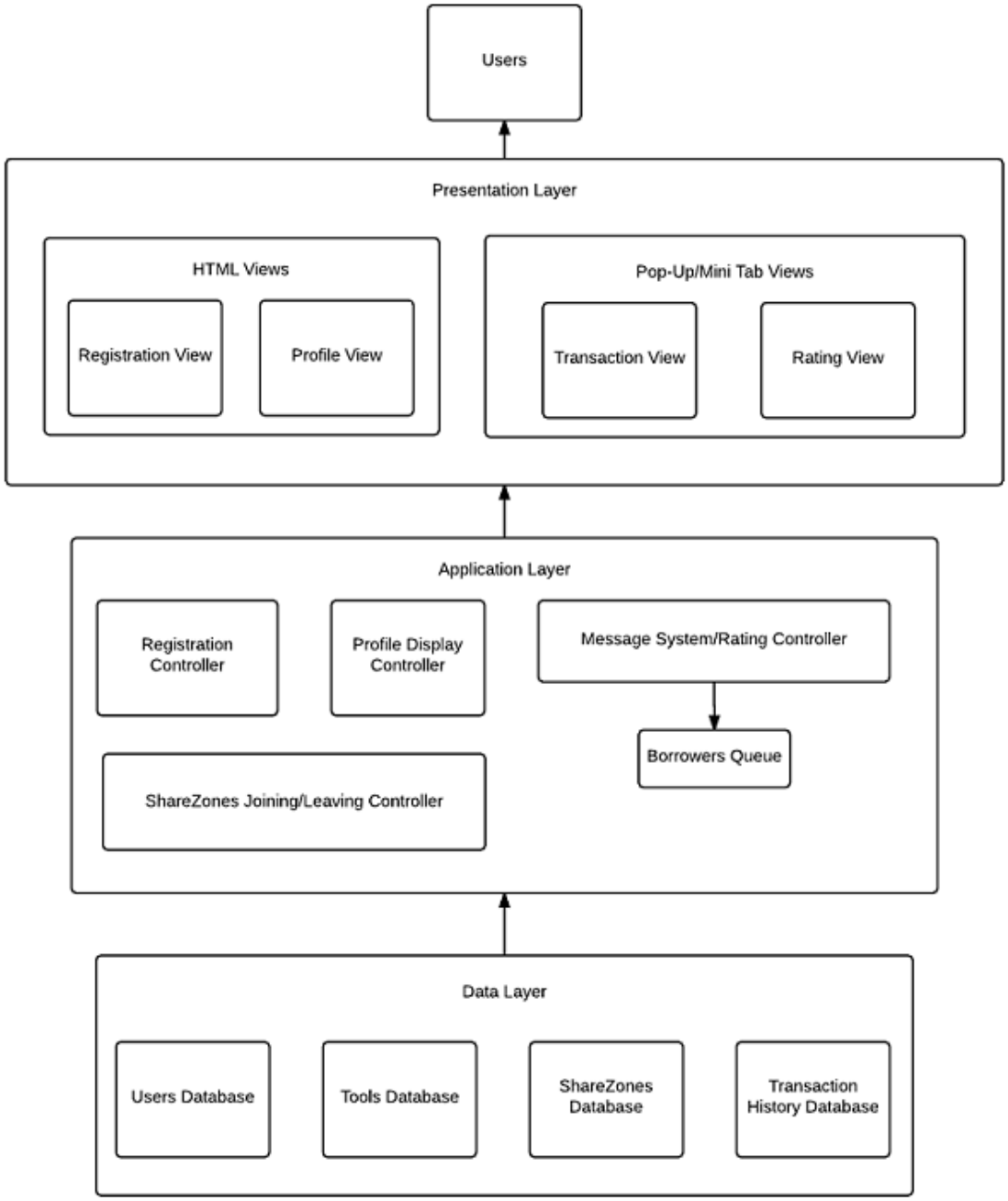
Product Design

Team: 261-03b, **Django Unchained**

Architectural Model

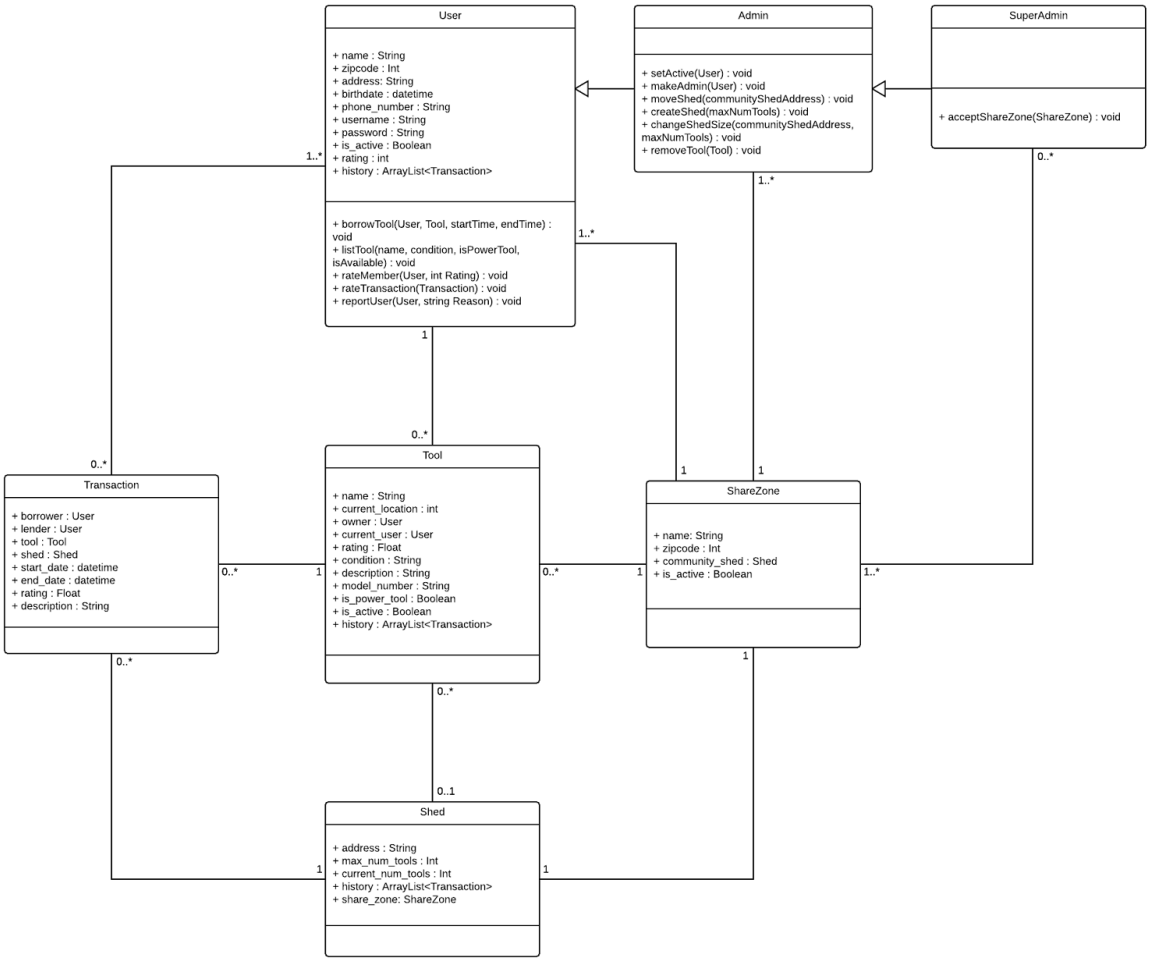


Components and Functions

Profile Database	Component State: Stores basic information for each profile such as: ● Name ● Zipcode/address ● Personalized Description ● Member Level ● What tools they own, and their starting conditions. ● User Rating (whether or not they take care of the tools they borrow) ● Rental History ● Whether they are banned or not Component Behavior: ● Updates user profile whenever other components require it. ● Gets ShareZone location from the ShareZone Database. ● Communicates with Tool Listing, ShareZone Database, Tool Return, Registration System, Admin Management, and Community Shed
Tool Listing	Component State: ● Stores data on the tools currently up for rent, and displays them for the ShareZone group they belong to. Component Behavior ● Updates the Profile Database by telling it which tools are being rented, or are up for rent, and where. ● Tells when a tool can be rented (where to pick up a tool). ● Tells who is renting a tool. ● Tells who the original tool owner is. ● Tells where the tool is currently ● Gets a description about the tools being rented from the original owner's profile database. ● Communicates with Community Shed, Profile Database, and Admin Management
ShareZone Database	Component State: ● Keeps track of the various ShareZones and their locations ● Keeps track of ShareZone owners (admins) ● Records who created each ShareZone ● Records which profiles are in each ShareZone Component Behavior: ● Provides the correct profile database for each ShareZone ● Tells the user if they are in the correct ShareZone ● Keeps and updates a list of current Admins of each ShareZone
Tool Return	Component State: ● Updates the availability for specific tools on the ShareZone Database as they are returned. ● Updates the Profile Database based on the addition of new owner and borrower ratings. Component Behavior:

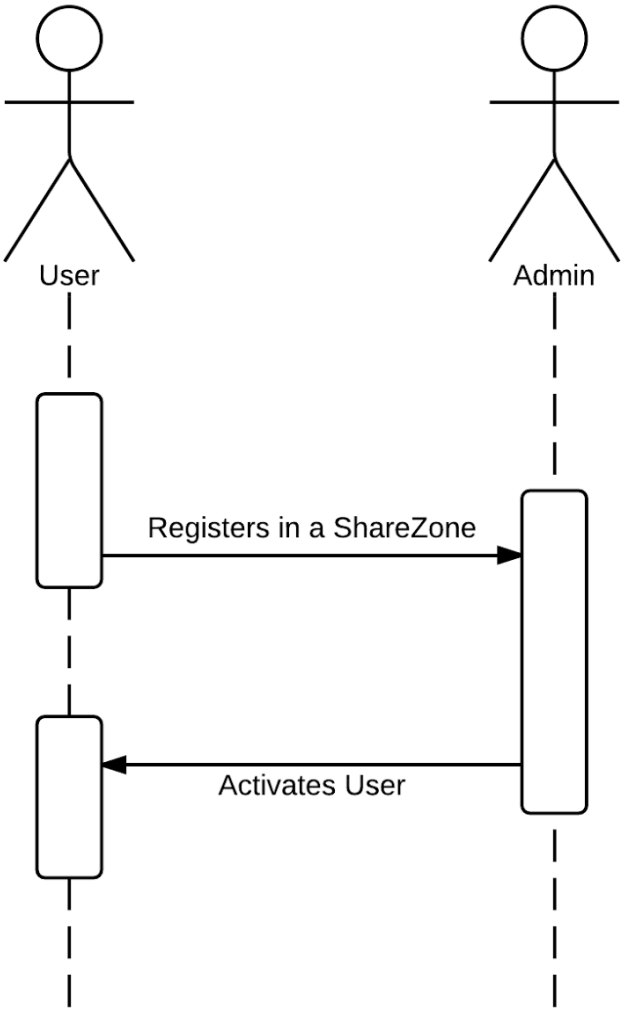
	● Updates the ShareZone database by changing the status of the tool to available upon successful return. ● Allows the user to submit a return condition of the tool, updates the tool listing. ● Allows the user to rate the transaction, rating is updated on the Profile Database. ● Allows the owner to accept or reject the specified condition of the returned tool. ● Allows owner to rate the transaction, rating is updated on the Profile Database. ● Allows owner to report the borrower if the tool is returned damaged, or not returned on time, borrower is flagged for review by admin.
Registration System	Component State: ● Creates new users ● Creates ShareZone if needed ● Users must be accepted by Admin Management Component Behavior: ● Gets information from ShareZone database to display a list of possible ShareZones to join (1 per member) ● Updates ShareZone database and Asks SuperAdmin if they can build a shareZone ● Updates Profile Database when a new user is created ● Asks Admin Management component to accept the new user into an admin's ShareZone ● Communicates with ShareZone Database,
Admin Management	Component State: ● This component will NOT keep track of whether a user is an admin. However, it WILL add more functionality for admin users. Component Behavior: ● Gets information from ShareZone database about whether a user is an admin for their ShareZone. ● If a user is an admin, they will be able to: ○ Choose or change the location of the community shed. ○ Promote members of their ShareZone to admin, which updates the profile and sharezone databases. ○ Allow members access
Community Shed (Database)	Component State: ● List of Tools currently stored in the shed ● Current community shed location. ● Description of community shed. ● Community Shed Admin Component Behavior: ● Gets information from Profile Database ● Gets information about its stored tools from Tool Listing Component

Class Diagram

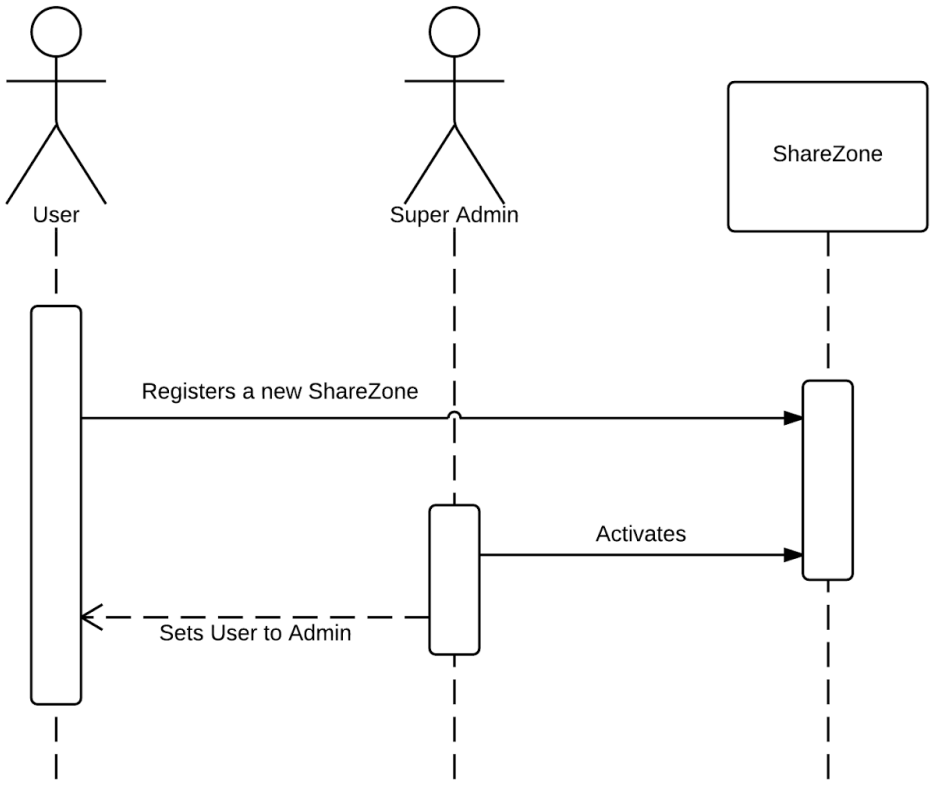


Sequence Diagram(s)

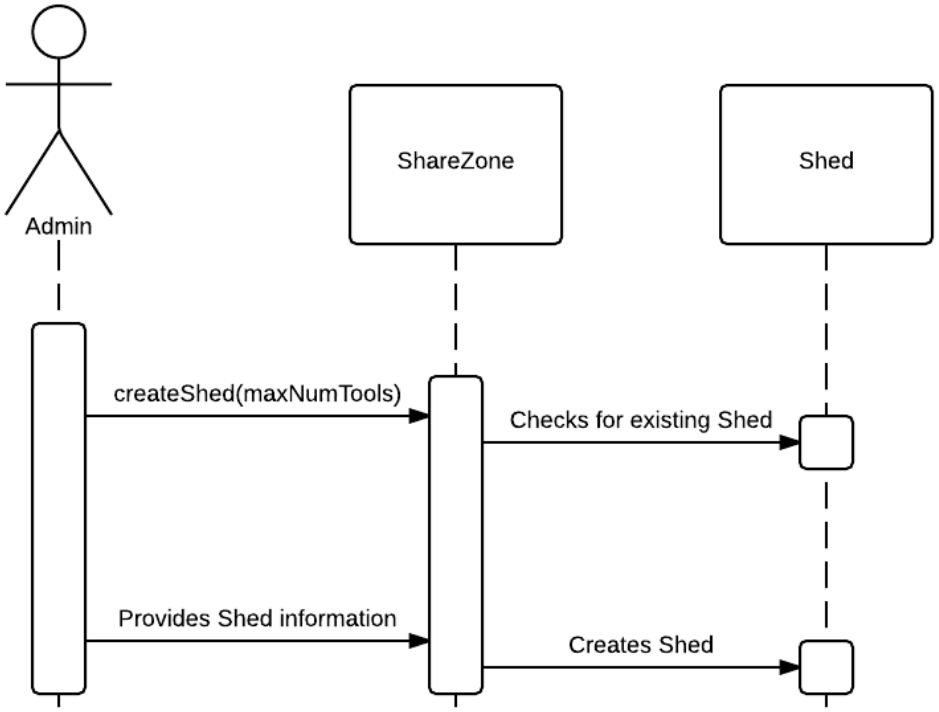
Creating a User



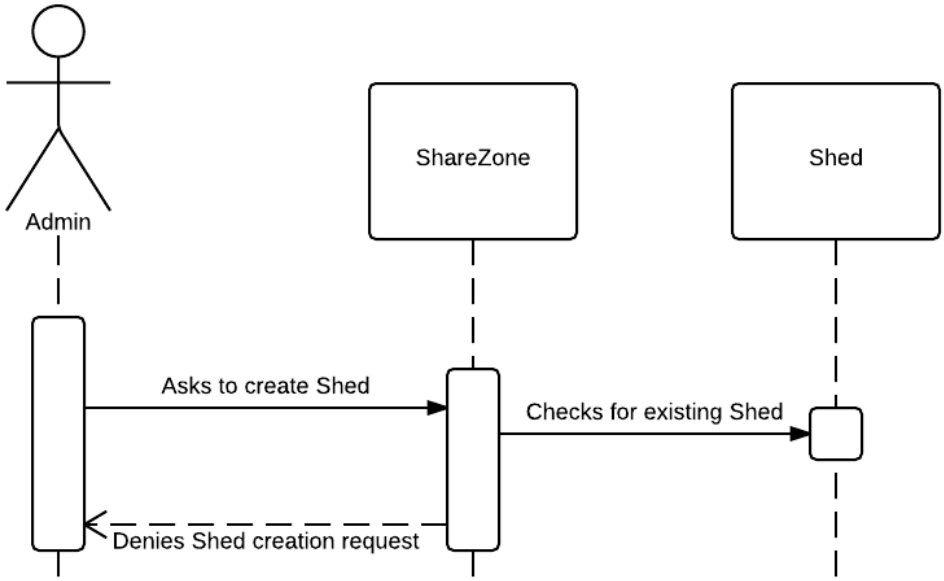
Creating a New ShareZone



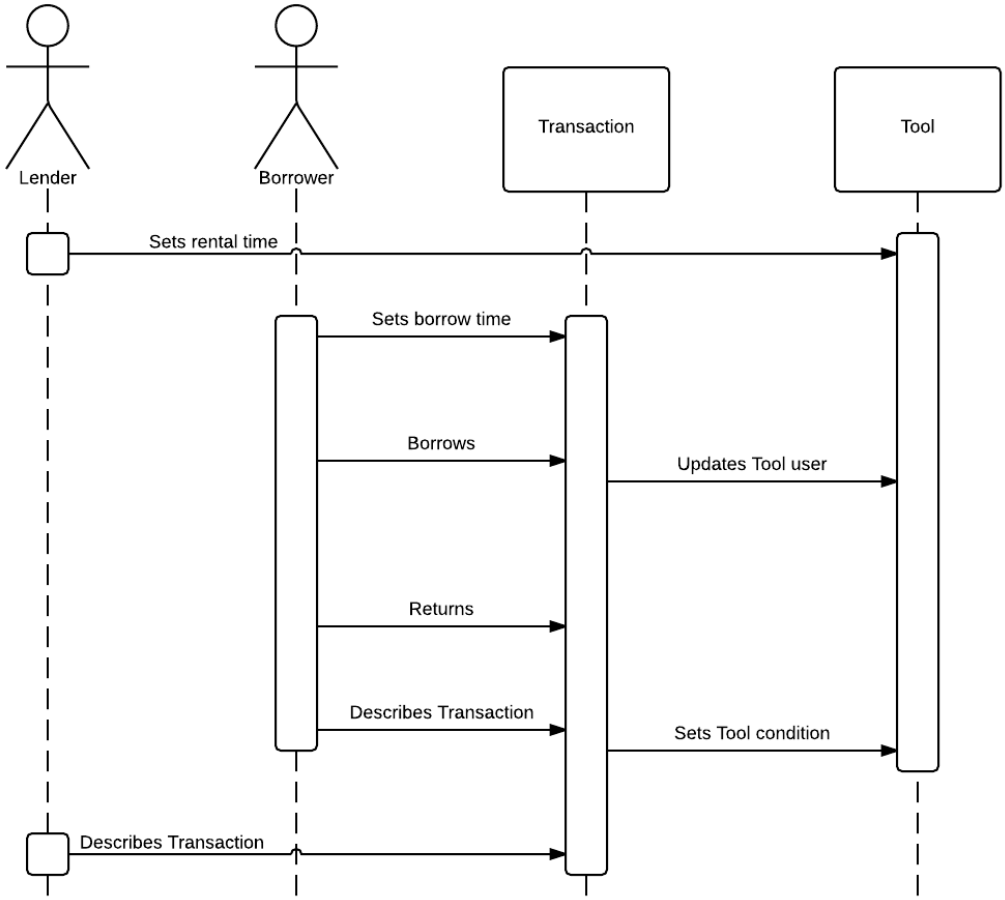
Creating a New Community Shed



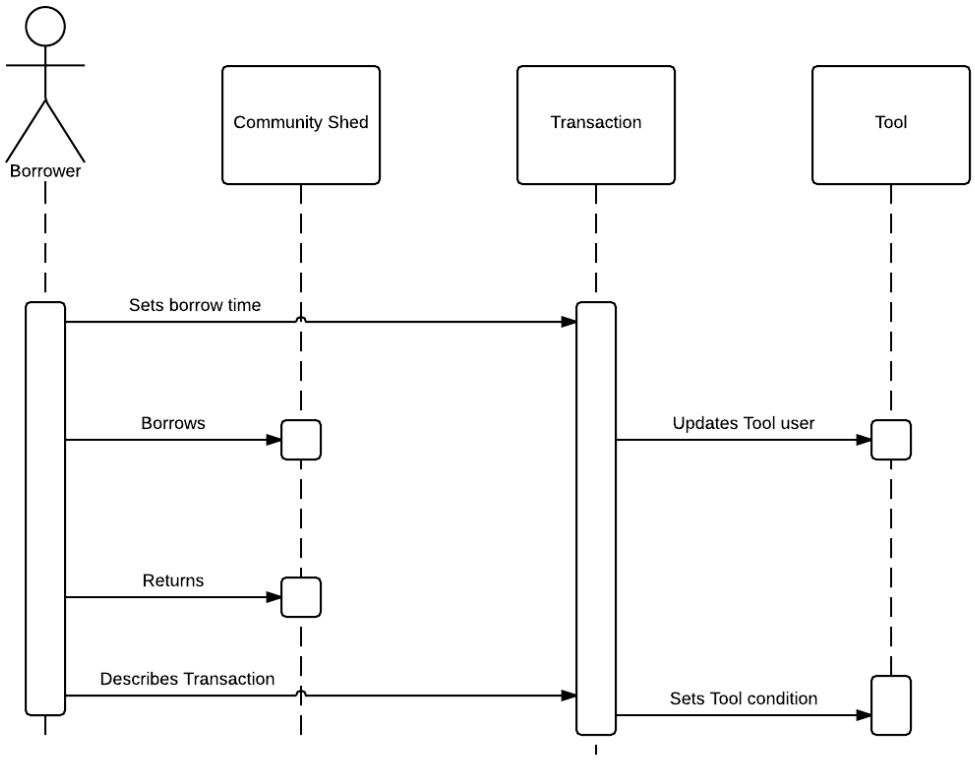
Failure to Create a New Community Shed



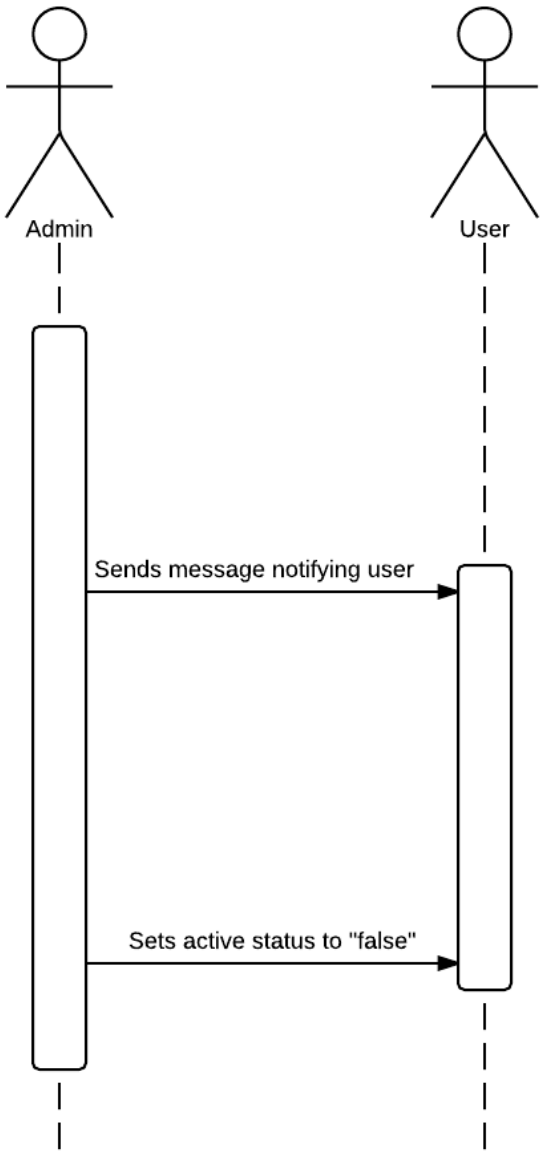
Tool Rental


Lender
Borrower
Transaction
Tool
Sets rental time
Sets borrow time
Borrows
Updates Tool user
Returns
Describes Transaction
Sets Tool condition
Describes Transaction

Tool Rental From a Community Shed



Deactivating a User



Design Rationale

What happens if the borrowed user gets deactivated? (what happens to their tools?)

The User object instance was first designed to be removed when their profile was banned. However, we now have established an isActive flag that allows a profile's data to be saved, so all borrowed items that the account still holds can be viewed and recovered, even when the account is inactive.

Issues with time (aging users and their birthdays)

The Users age was originally kept using an integer value. The integer value kept track of their age, but was difficult to update for when the user's birthday occurred. Therefore, a datetime variable called "age" was added, which will be able to keep track of both their age and birthday, so updating the user's actual age will be much easier.

Tools no longer have a rental countdown time, now they are given a start date and end date

A counting variable was first used to keep track of the amount of time a user had to rent a tool. Since the system is designed to be static and only change when it needs to, the counting variable was inefficient. Instead, two datetime variables named rentalStartDate and rentalEndDate were used to keep track of the renting process, without having to change values at any time.

Removed most of the update functionality for a user class.

The system flow as a whole was far too dynamic to efficiently run with large amounts of data. A more static system was developed to make changing the information less frequent. Our mentality changed from one focused on keeping track of every change in every class, to one that keeps track of what has been added, but does not focus on changing that information.

Owners and current users for tools are no longer String based, now they reference an actual User class instance.

The tools were originally designed to reference a username. However, this method would require more information taken from the database. To fix this issue, the tool's owner and current user will reference a specific User class instance, which makes collecting info from the instance very simple, without requiring any searching.

Rating system and tool condition have been switched from String values to enumerated integer values for increased adaptability.

The condition of a tool was originally designed to be a description of what the tool's condition is. However, enumerating and integer value for the condition of a tool provides a standardized range of conditions (for example, 0 would relate to "Broken" and 5 would relate to "New").

Zipcode has been switched from Integer values to Strings.

This change was made so that the leading zeros on zipcodes would not be removed. Originally we used integer values with a limit of five characters for zipcodes but after testing different values we saw that the zeros at the front of these zipcodes would not be there when they were put in the db. We fixed this by making all zipcodes be strings and limiting the user so they can only type in numbers.

Handling the transactions has been separated into its own class.

The handling or transaction reporting and storage was split among multiple classes. To simplify the design, a transaction class was created to keep track of an individual transaction between two users, or a user and a community shed. This allows many pieces of information needed for a single transaction to be congregated into a single block of information that can be kept as a records of rentals in a community.

ShareZones are no longer displayed solely on location.

We changed ShareZones so that all of them will be displayed to the user regardless of where they are. Originally we were going to have it so that you could only see the ShareZones around you, but we found it easier to just let the user choose, while still showing them the ShareZone's zipcode.