Revision History

Version	Date	Comment
1.0	1/26/2018	Initial Release
1.1	2/5/2018	-Added unique username to admin class (just like visitors and owners have) -Visitors must leave a 1-5 rating if they decide to log a visit -Tables that show confirmed properties should have an "Average Rating" column that shows the average rating of all logged visits -The average rating column in every table must be sortable -The detailed view of a confirmed property shows the average rating as well
1.2	4/9/2018	-Visitor table should be sortable by visits as well

Atlanta Gardens, Farms, and Orchards

Spring 2018 - CS 4400 Database Project
Serve-Learn-Sustain

Project Purpose

Analyze, specify, design, implement, document and demonstrate an online system. You are required to use the classical methodology for database development. The system will be implemented using a relational DBMS that supports standard SQL queries. The TAs will provide you with information about how to access a college-managed MySQL server in order to implement your database and the application. The professors must approve any other alternative implementations. ***In no circumstances can you use a tool that automatically generates SQL or automatically maps programming objects into the database. You also cannot use any other software like Access or SQLite.*** Ask the professors or TAs if you have questions about which tools/languages/software are allowed.

Project Phases

The three phases of the project cover the following work-processes from the Classical Methodology for Database Development (see notes on T-Square under resources). Slides on database design methodology will be useful for phases 1 and 2: All slides have been posted on T-Square.

Re-grade Policy

Once graded phases and/or quizzes are returned, there is a one-week deadline during which you can contest your grade with the TA who graded your assignment. This clock starts not when you personally get your returned paper, but when the papers are returned to the class.

Teams

Project teams consist of 4 or 5 members. You are allowed to form teams across the two sections (A & B) of the class. A team may remove a team member from further participation in the team when Phase I is turned in or when Phase II is turned in. A written notification with a proper justification must be provided to the professor and the Head TA at that time in hard- copy form.

Deliverables

Phase 1 - Submitted to T-Square and bring hard copy to class

The deliverables include (in a single PDF file):

1. A cover page - you **MUST** include all information listed on the template (See T-Square Project Folder).
2. Enhanced Entity Relationship (EER) Diagram
3. Information Flow Diagram (IFD)
4. A list of **logical** constraints. You are required to include at least **three (3)** constraints, although a fully specified system will probably have more than that.
5. Any assumptions made, with justification and explanation.

Additional Phase 1 Information:

1. The EER must capture the functionalities of the application system whenever applicable. (e.g., total participation, superclasses/subclasses, weak entities)
2. The design of your system must include all functionalities as indicated by the application description in this document. You are allowed to make up additional assumptions as long as they do not conflict with the specified constraints and requirements. You must list all your assumptions; otherwise, your EER diagram will lose points since the TA will not understand certain parts of your design based on certain assumptions you made. Information extraneous to the required functionality may count against you.
3. Logical constraints that can be specified directly using ER notation will not count towards the three required. Constraints related to data type or values are not accepted as constraints.

Each team needs to turn in one hard copy (only 1 per team). Every student must upload an electronic copy to T-Square individually. You will receive a -5 points penalty if you do not submit an electronic copy. **Please write your team number clearly on the cover page.** If you do not know your team number, email the Head TA.

Phase 2 - Submitted to T-Square and bring hard copy to class

The deliverables include (in a single PDF file):

1. A cover page, same as Phase 1.
2. Copy of the EER diagram (either your phase 1 diagram, with any modifications, or the provided solution)
3. Relational Schema Diagram - identify primary and foreign keys and show referential integrity using arrows.
4. MySQL CREATE TABLE statements, including domain constraints, integrity constraints, primary keys, foreign keys, & appropriate referential triggered action clause.

Each team needs to turn in one hard copy (only one for the entire team). Every student must upload an electronic copy to T-Square individually. You will receive a

-5 points penalty if you do not submit an electronic copy. **Please write your team number on the cover page.** If you do not know your number, email the Head TA.

Phase 3 - Submitted to T-Square

The electronic deliverables include:

1. A cover page, same as Phase 1 and Phase 2.
2. A text file with all SQL statements for each task. (Follow the template in the Phase 2 design methodology).
    1. NOTE: A set of SQL statements may be required in order to complete one task. However, in such cases, the last SQL statement should show the output according to the specification. Views and nested queries may be used to support the tasks.
3. For the heavy weight project option, your source code for the application.

Notes:

1. Prior to the demo, the TAs will give guidelines for populating the database with data. The database has to be populated with this data set prior to the demo.
2. **Every student must upload an electronic copy to T-Square individually. You will receive -5 points penalty if you do not submit an electronic copy.**

On Demo Day

Bring your laptop and make sure you have a text file on your laptop with all of your SQL queries just in case your application does not work. More details about demos will be discussed later this semester. **All team members must be present and on time.** Missing/late team members will receive a -10 points penalty.

Grading

The project consists of three phases (deliverables) as well as a final demo to the TA.

Phase 1 and Phase 2: 10% of your final grade

Phase 3:

Heavy Weight option (20%): Your team will use the embedded SQL feature of MySQL, which allows you to embed SQL statements in a standalone application.

Light Weight option (5%): Your team will demo the SQL queries on the MySQL console. Your team will also be required to take the final exam.

NOTE: You can always change your project option until the demo starts. Once the TA has begun demoing your application, you cannot change from Heavy Weight to Light Weight (or vice versa).

Final Exam (15%): This is only for the students who opt for the Light Weight option. Students who opt for the Heavy Weight option **cannot** take the final exam.

Project

For this project, you will create an application system or a tool that stores location, usage, and ownership information on gardens, farms, and orchards in the Atlanta area.

The following sections contain a functional description of the system along with some mockup screenshots. Each section explains functionality and then presents an example screen about it. **You don't have to follow the UI designs, but your program needs to support all of the functionalities. (Pay close attention to tables – they have arrows to indicate which columns/attributes are sortable and searchable. Buttons are gray or have an arrow to indicate that it should be a drop down menu/radio button).**

Searching should be an option to reduce the size of any table. Depending on the table, the search parameters will be different (different tables have different columns). You may have a drop down or some other way of specifying which column you are performing the search on.

These mockups are just for helping you to understand all of the functionalities. **You will not be graded on how similar your UI looks to these mockups, but you will be graded on how well everything works (heavyweight).** A complete reorganization of the user interface is permissible (and encouraged!) as long as your application supports all the functionality listed below. The sections have been grouped by registration, owner, visitor, and administrator functionalities.

For the Heavyweight option, you may implement the project as a traditional standalone application (e.g. using Java GUIs) or as a web application (e.g. using a web scripting language like PHP). There is a list on T-Square about which languages/tools/software/platforms are allowed. A Piazza post will be maintained where you may ask if certain technologies are allowed**. If you do the heavyweight option, we reserve the right to deduct points on flaws with the UI design not included in the discussed functionalities (things such as having to manually reload a page to update the tables, having to re-enter information for all attributes when only updating a single attribute, not having navigation/back buttons, etc.)**

User Accounts

Log In

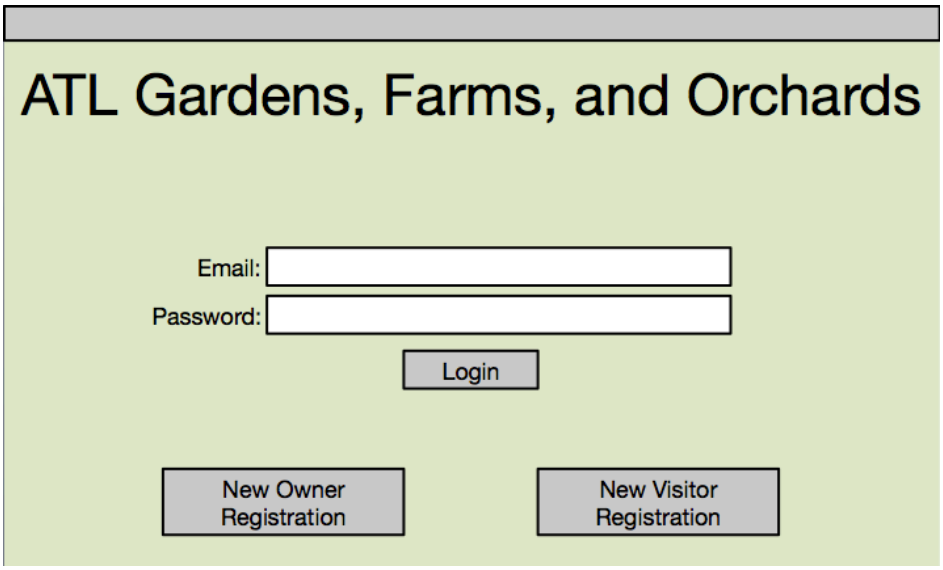
A user must log in before using the application. There are three types of users – administrators, owners, and visitors. To log in, a valid email and password combination is required.

Brief overview of all account types:
Owners: add property information to the database, can modify their own properties and view other owners' property information
Admins: verify that the information added by the owners is correct/valid, can view and modify (including delete) property information, validate and add additional plants/animals, and delete owner/visitor accounts
Visitors: provide visitation metrics for validated properties by logging their visits, can view public, verified properties and modify their visitation on them

If a user provides invalid login credentials, an error message should be shown on the screen.
If a user does not have an account, they can click on the appropriate register button to create an account that meets their needs.

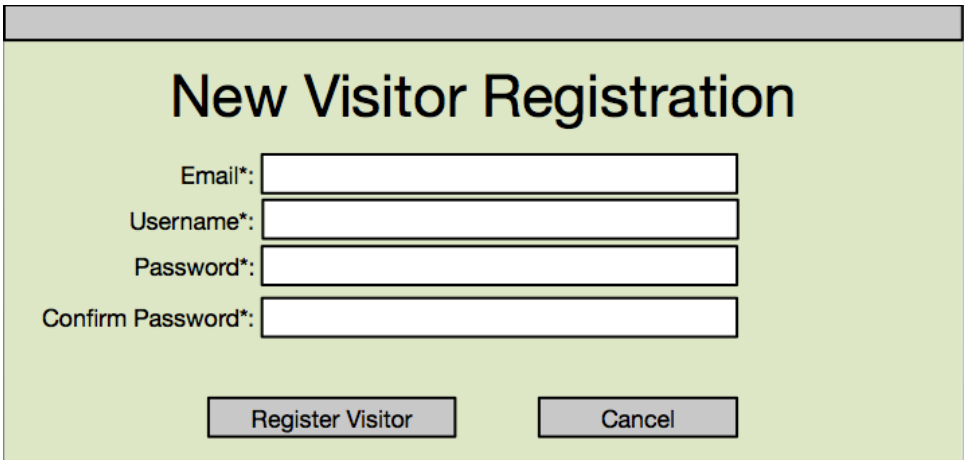
Notes:

- Email and username are **unique** for each user (individually unique, not the combination).
- Since all user types share the same login screen, you will need to determine what the user type is before proceeding to the next screen.
- Visitors and owners are the only types of accounts that can be registered (admin accounts will be pre-populated into the database)



New Visitor Registration

After clicking the Register New Visitor button on the Log In screen, the user will be redirected to a registration page to create a visitor account. All fields are required.



Notes regarding registration in general

- Email must be **unique**
- Username must be **unique**
- Password and confirm password must match
- Passwords must have at least 8 characters
- You should not store the password as a string in the database. Instead, run the password through some sort of hash function (most languages have a built in hash function that you can use) and store the result. When a user logs in, you will use the same hash function to verify the account.
- Administrator accounts are created in the database – the create account page can only be user to create owner and visitor accounts.
- You may validate the email address using the database (SQL CHECK constraint) or in the front-end application.
- ~~Only owners and visitors have usernames.~~
- Owners, visitors, and admins have **unique** usernames
- An email address consists of alphanumeric characters, followed by an @ symbol, followed by alphanumeric characters, followed by a . symbol, followed by alphanumeric characters.

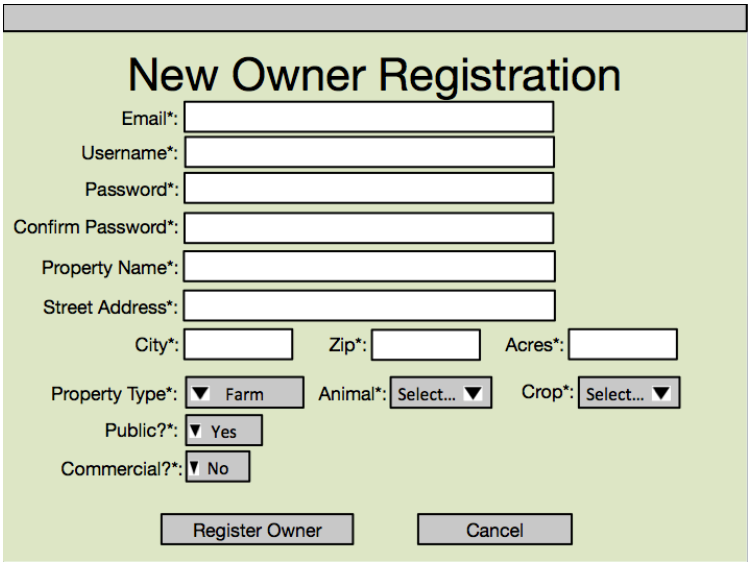
New Owner Registration

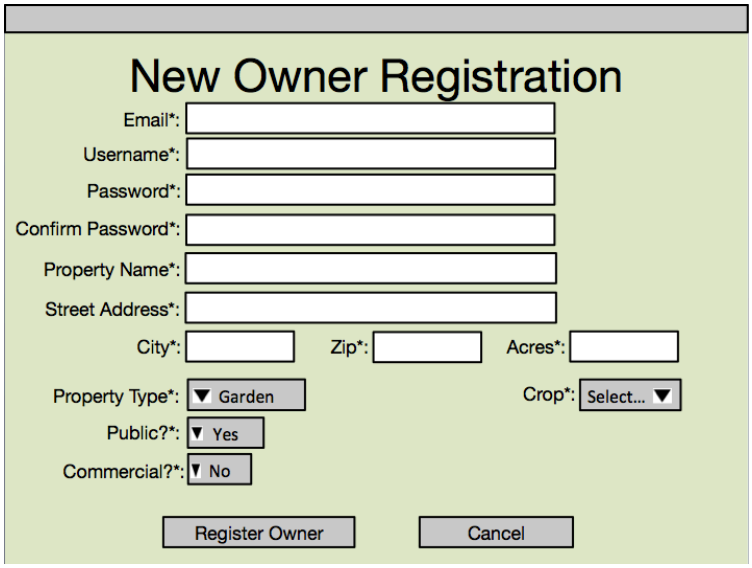
After clicking the Register New Owner button on the Log In screen, the user will be redirected to a registration page to create an owner account. All fields are required. Requirements with regard to email, username, and password will match those of the visitor accounts.

Requirements (If not met, display error message)

- Owners must specify one property (i.e. garden, farm, or orchard) upon account creation. HOWEVER**, if all of their properties are removed, their account remains intact**.
- Upon successful account creation, the property specified will also be created and will be given a **unique numerical 5-digit** property ID. Until the property is validated, it's initial status will be **invalidated**
- Depending on the type of property, different fields will be required (farms require animal and crop, garden and orchard only require a crop). Further explanation on properties is given in the "Add Property" section below
- When registering a new owner account, the owner must select an animal/crop from the approved list. To add an animal/crop that is not yet approved, they must go through the "manage property" page as detailed below
- Owner account creation will require listing only ONE initial crop/animal for the property (depending on the property, only farms have animals), more crops/animals can later be added in the manage property screen
- If the owner wants to list an animal or crop that is not present in the dropdown list for the category they chose, they must submit the animal/crop for approval through the manage property page

Notice the difference in the below screens: on the left, a farm is selected as the type, so there is a dropdown for both an animal and a crop. Note that initially we allow only one animal and one crop. On the right, property type is garden, so there is only a crop dropdown. The dropdowns should be populated with animals/crops that are allowed on this particular kind of property (detailed description in the "Add Property" section).

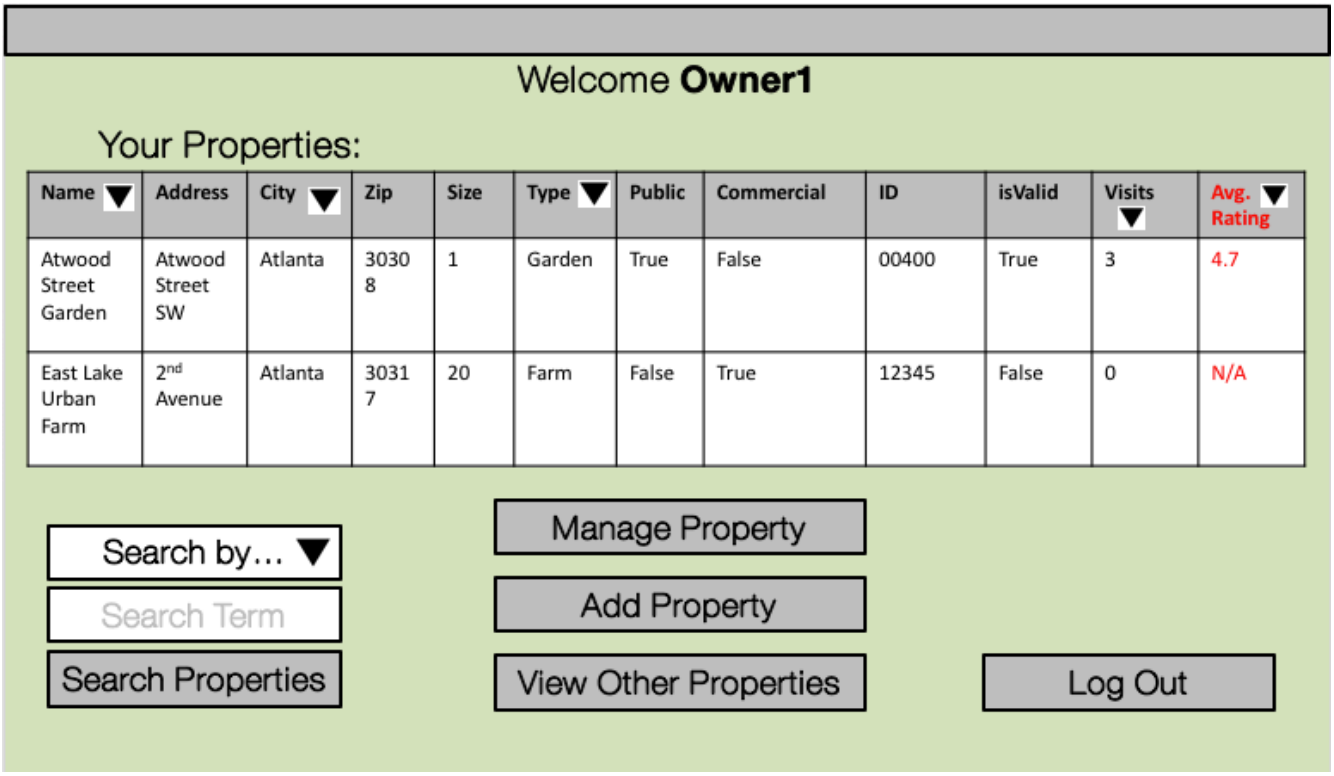




Owner Functionality

When an owner logs in, the default view should be a table listing all of the owner's properties. They should be presented with the following owner options:

- Add a new property
- View other owner's properties
- Manage a property
- Search their properties
- Log out



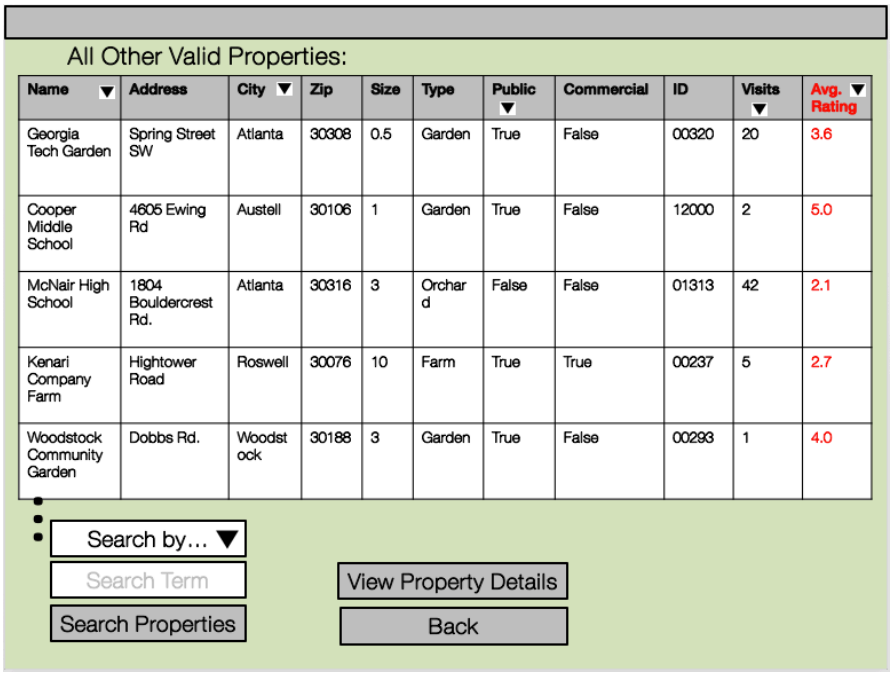
To manage a property, it should be selected from the table of available properties. Note that an owner who doesn't have property should have an empty table and not be able to manage any properties.

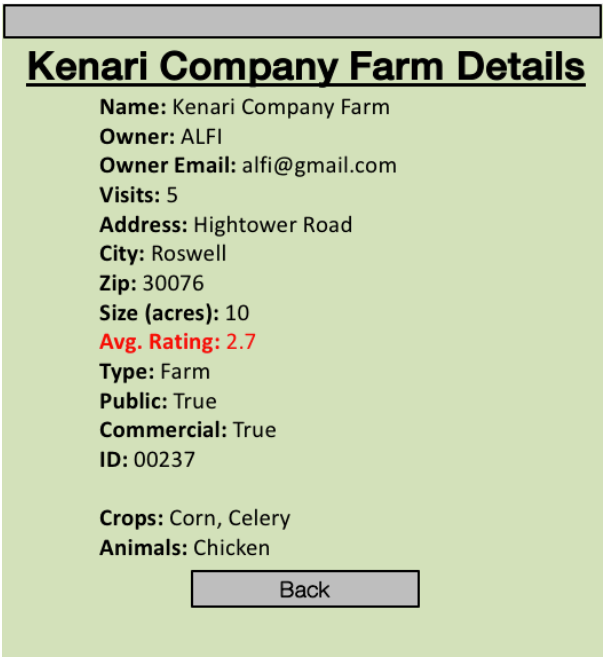
View Other Owners' Properties

If the owner clicks view other properties, they should be presented with a table that contains all properties that belong to other owners that have been confirmed by an administrator (**not including** their own properties). The table should contain all relevant information regarding a property. By default sort alphabetically. The table shown has arrows on the columns that should be sortable.

Notes:

- The owner should be able to sort this list by various attributes/columns.
- The owner should be able to select a property from the list to view all details about the property.
- The owner should be able to narrow the list by performing a search





Manage Properties

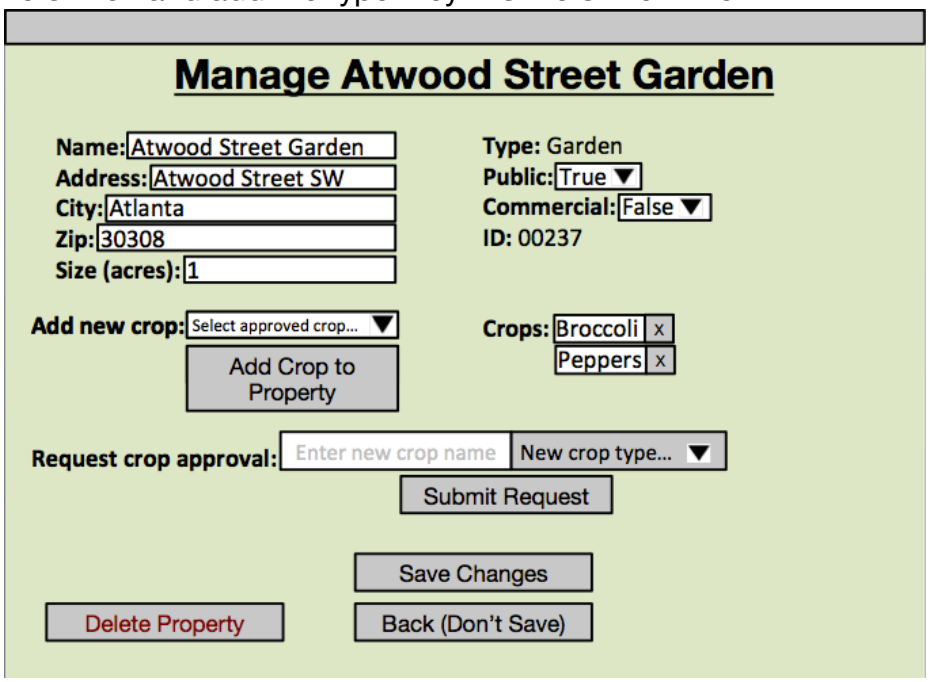
The owner should be able to select one of their properties to manage. The manage property screen should allow the following functionality:

- Add a new animal/crop to the property from the approved list
- Request to add an animal/crop to the approved list (crop has name and type)
- Remove an existing animal/crop (unless it's the last one listed). Remember a property must maintain at least 1 of everything required to fit its classification.
- Increase/decrease the size of the property
- Modify if is it public or private
- Modify if it is commercial or non-commercial
- Modify the name (name must remain **unique** to the system)
- Modify the address
- Remove the property entirely

When an owner modifies their confirmed property, the status of the property should be switched to **unconfirmed/invalidated** and will require re-confirmation in order for other owners and visitors to view it. If the property was already unconfirmed, it just remains in that state. If an owner attempts to save a modification that is invalid, **discard the entire modification or show an error message,** i.e. do not partially update something if a portion of it is incorrect, display an error message instead.

Notes:

- A property should not be able to switch types. i.e. an owner cannot change a farm to a garden. To achieve this, the owner should remove the property they wish to switch and add the type they wish to switch it to.



Notice: the above screen has all of its fields pre-populated because it is being edited as opposed to being entered in the system for the first time.

Add Property

If the owner clicks add property, they should be taken to a form to fill out the details of the property they wish to add. A property must be specified as a farm, garden, or orchard. A property must be associated with an owner account.

For the purpose of this project, please make the following assumptions with regard to the difference between the three types of property:

- Farms **must** have at least 1 animal and at least 1 crop. Farms can grow any crop that they want from any of the 4 potential crop categories, which are: fruits, vegetables, nuts, and flowers.
- Orchards are used only for growing fruits and nuts. They are not allowed to grow vegetables or flowers. They are not allowed to have animals.
- Gardens are used only for growing vegetables and flowers. They are not allowed to grow fruits or nuts or raise animals.

Creating a type of property must specify 1 of each thing required to classify the property. i.e. creating a farm must specify 1 animal AND 1 crop (fruit or vegetable or nut or flower), creating a garden must specify 1 flower or 1 vegetable, and creating an orchard must specify 1 fruit or 1 nut. More crops that are valid for the property type can be added later from the manage property screen. When adding a new property, the owner MUST specify a crop/animal from the approved list. Animals/crops not on this list can be requested from the manage property page as shown on the previous page.

Other things that must be specified when adding a property:

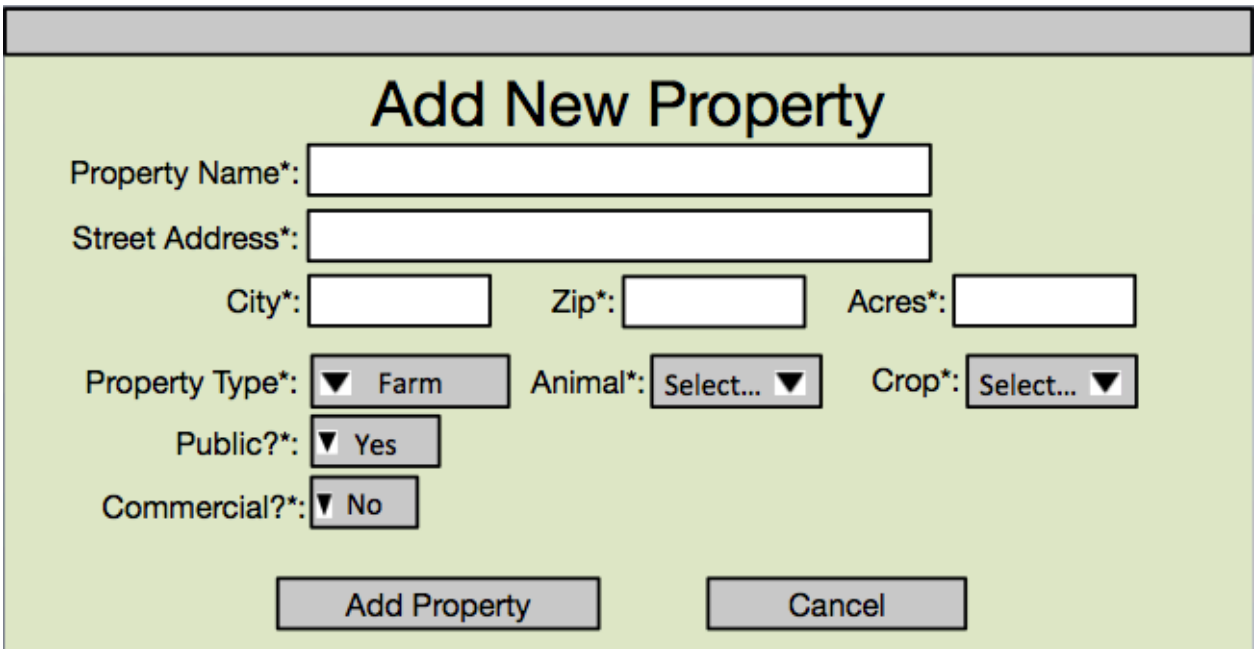
- The name of the property (must be **unique**)
- The address of the property (street, city, zip)
- Size in acres
- Whether it is commercial or non-commercial
- Whether it is private or public

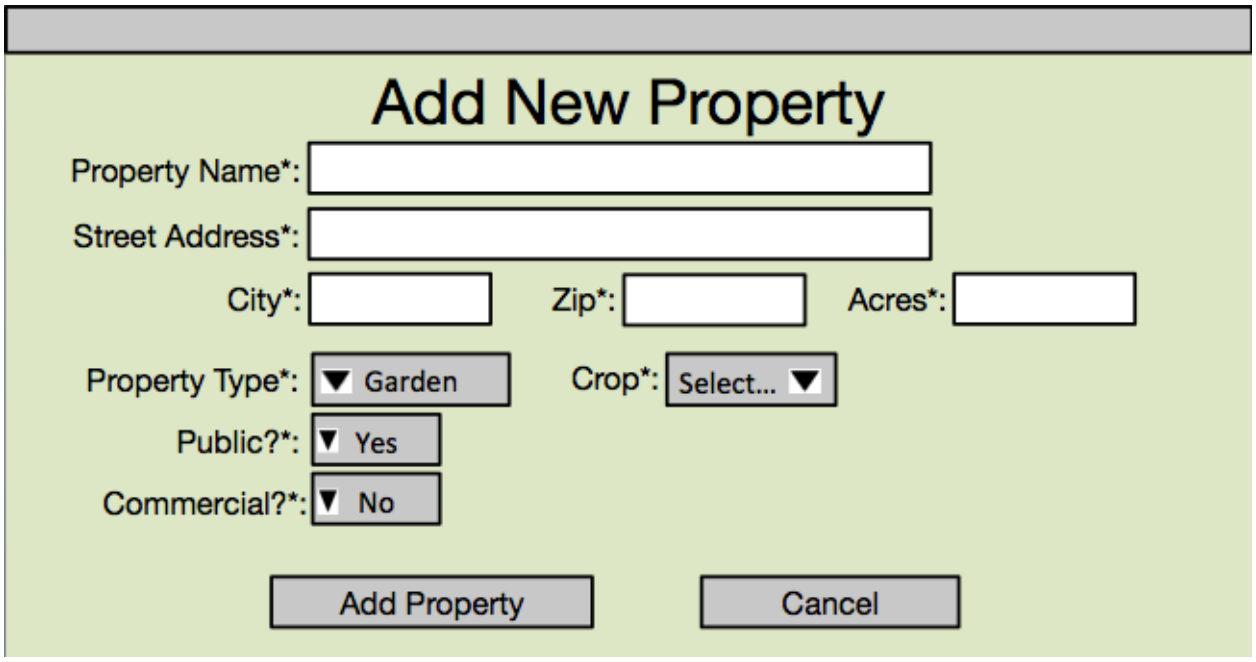
By default, an added property will be **unconfirmed/invalidated**. An admin must confirm/validate the property before other owners and visitors can view it. An added property should be automatically assigned a **unique numeric 5-digit** property ID by the system (yes, this system assumes that there are less than 100,000 such properties in the Atlanta area).

Notes:

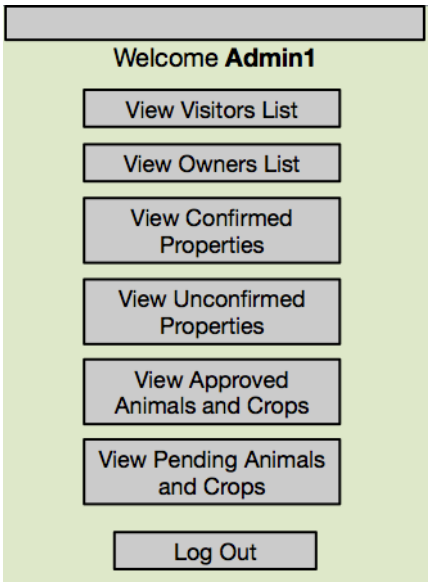
- It is possible for a garden, farm, and orchard to all have the same address and owner (or even different owners). In such cases, it is only necessary that each property maintain the minimum amount of information required to fulfill its classification.
- If the same property grows all 4 types of crops but has no animals (i.e. it's not a farm), consider listing the property twice: once as an orchard and once as a

garden. Remember, only name and property ID must be unique attributes, while address need not be.





Notice: there is a difference in the above screens. One is attempting to add a farm and the other is attempting to add a garden. When adding a new property, the owner MUST specify a crop/animal from the approved list. Animals/crops not on this list can be requested from the manage property page.



Administrator Functionality

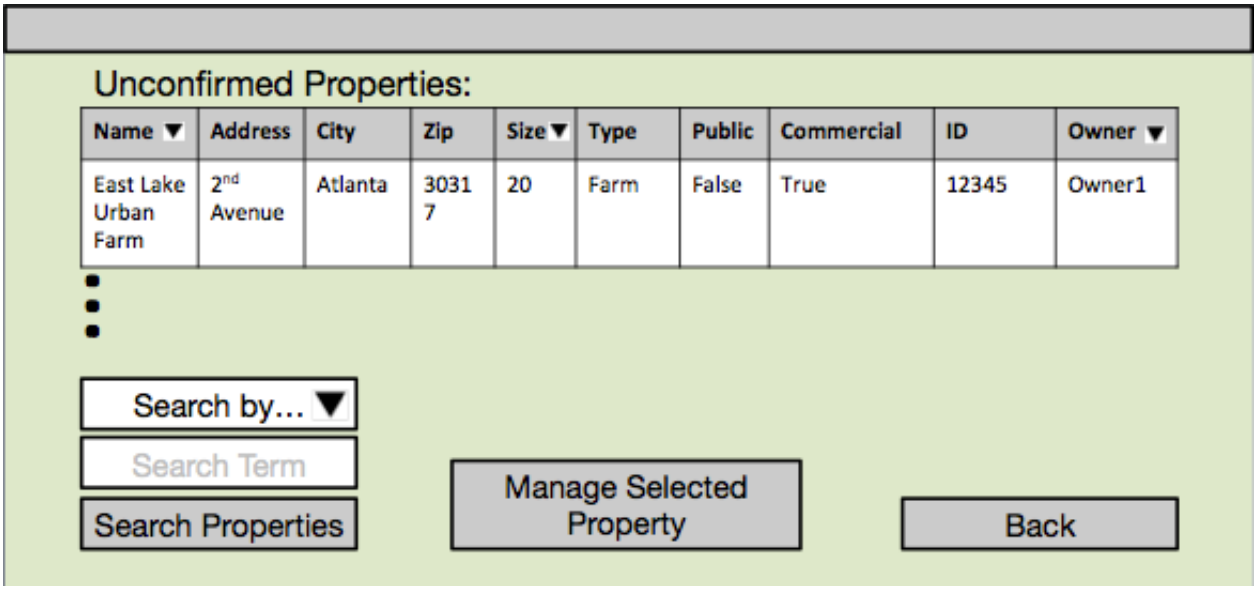
When an administrator logs in, they should be presented with the following admin options:

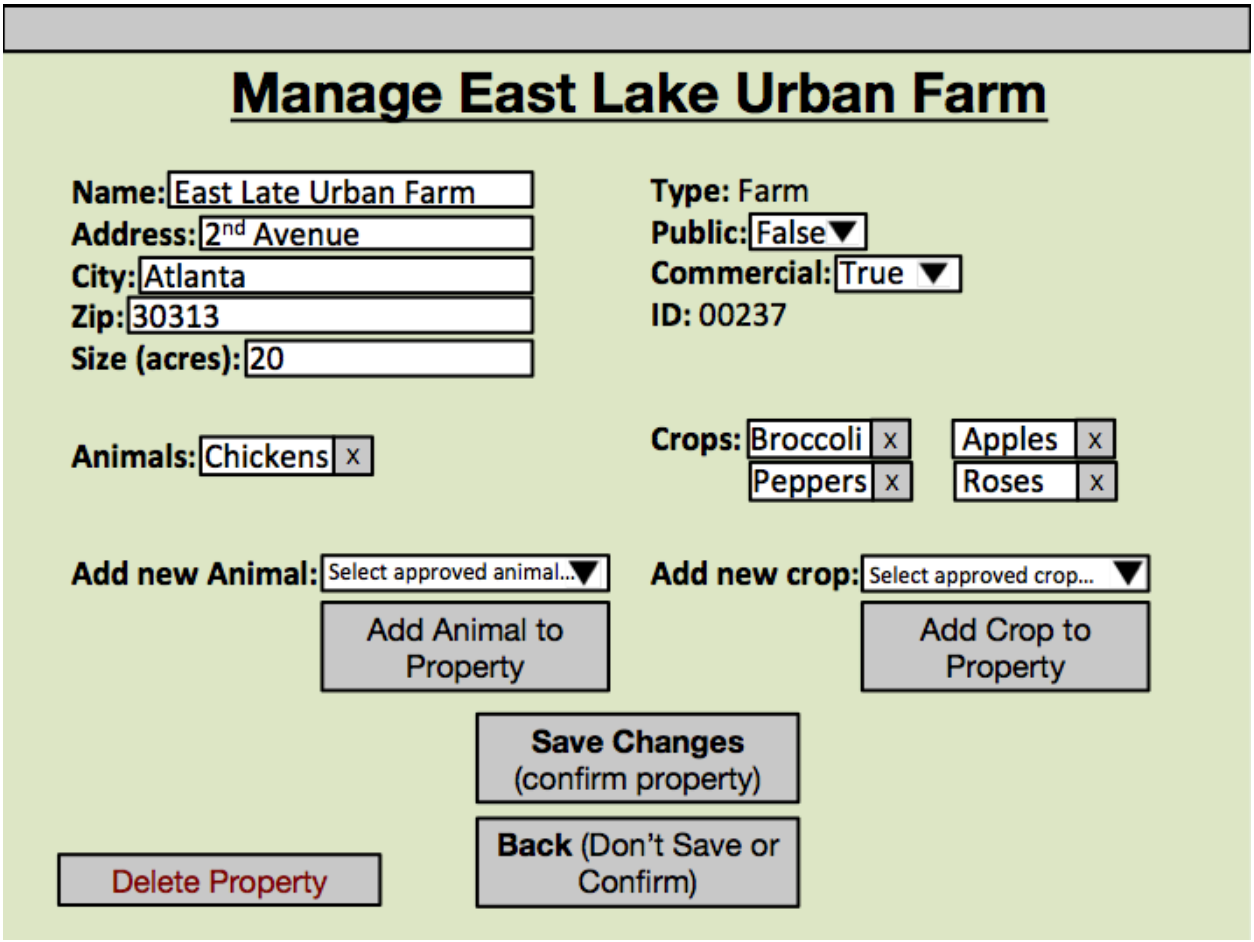
- View list of all visitors
- View a list of all owners
- View all confirmed properties
- View all unconfirmed properties
- View all approved animals/crops
- View all pending animals/crops
- Log out

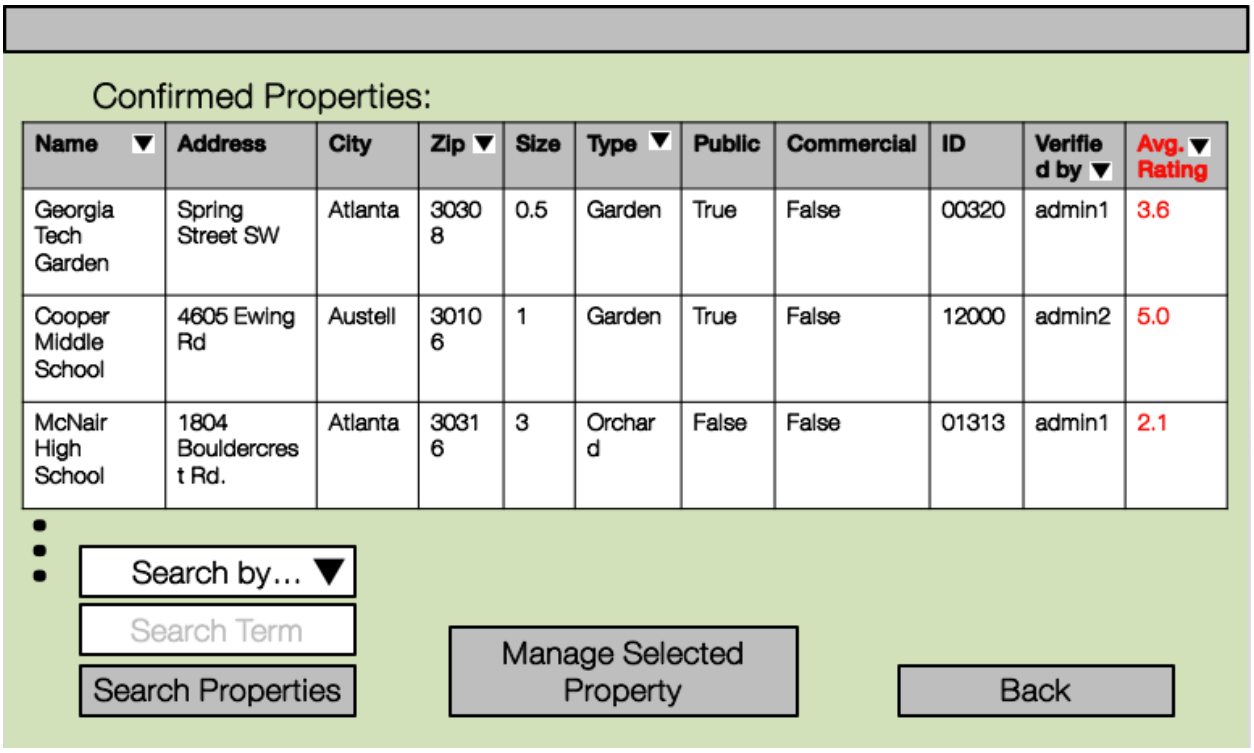
If the admin selects either “View Confirmed Properties” or “View Unconfirmed Properties,” the following functionalities should be the same: the admin can select a property entry from the table to view more details about/manage the property. They should be taken to a screen that is similar to the owner’s manage property screen. From there, they should have all of the same functionality as an owner, namely:

- Add a new animal/crop to a particular property from the approved list
- Remove an existing animal/crop from the property (unless it’s the last one)
- Increase/decrease the size of the property
- Modify if is it public or private
- Modify if it is commercial or non-commercial
- Modify the name (name must remain unique)
- Modify the address
- Remove the property entirely

The difference with these modifications is that when they are made, the properties will remain in or be switched to the **confirmed/validated** state so that other owners and visitors can view them.



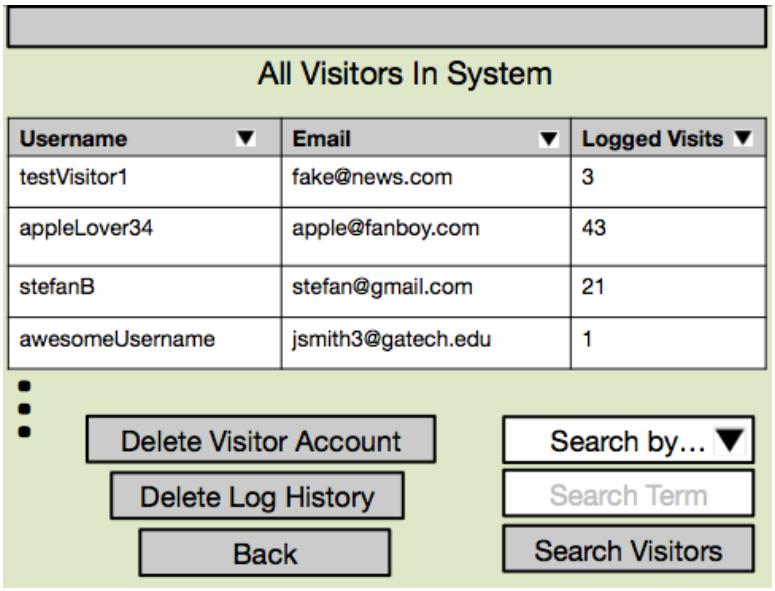


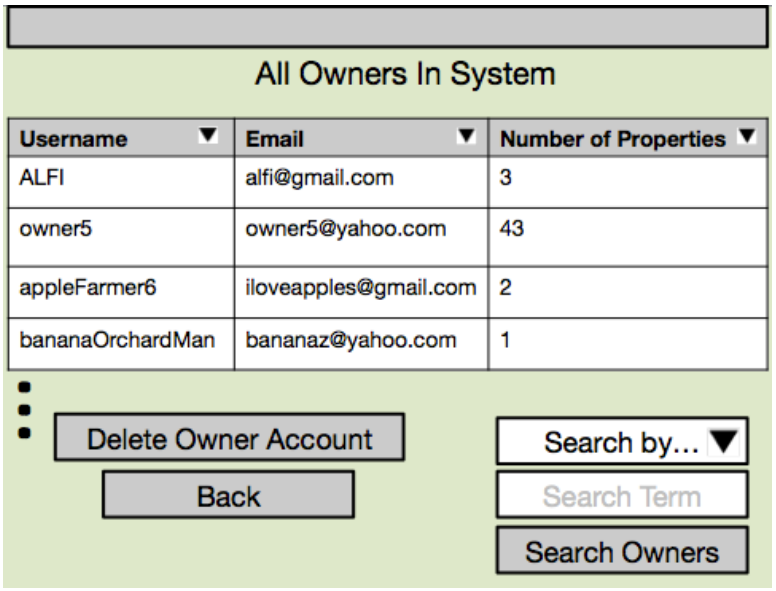


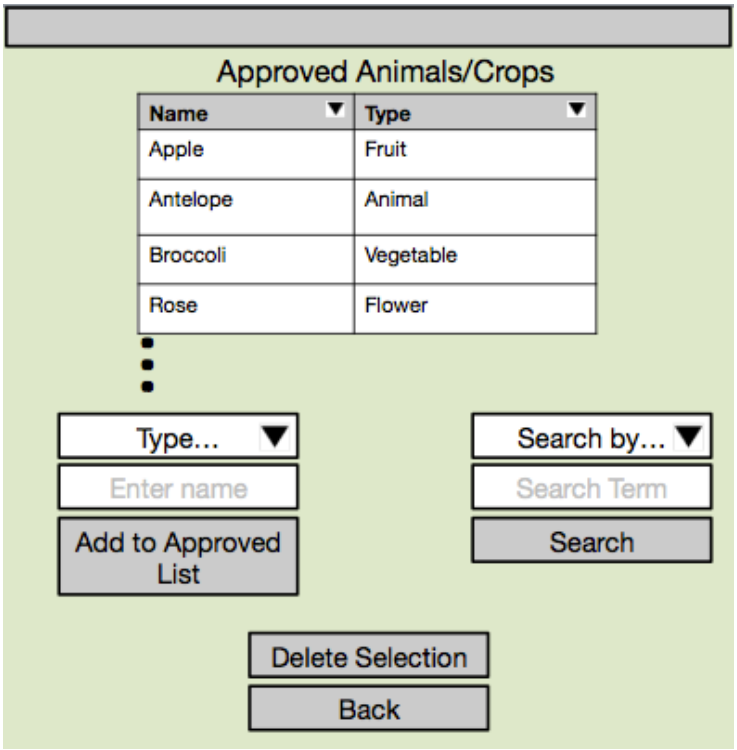
Name	Address	City	Zip	Size	Type	Public	Commercial	ID	Verified by	Avg. Rating
Georgia Tech Garden	Spring Street SW	Atlanta	30308	0.5	Garden	True	False	00320	admin1	3.6
Cooper Middle School	4605 Ewing Rd	Austell	30106	1	Garden	True	False	12000	admin2	5.0
McNair High School	1804 Bouldercrest Rd.	Atlanta	30316	3	Orchard	False	False	01313	admin1	2.1

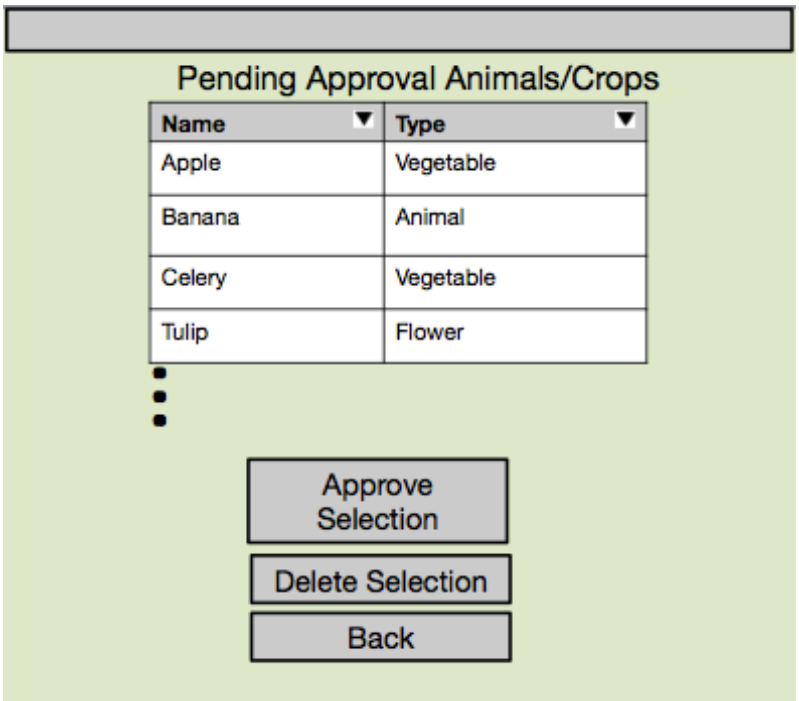
An admin should also be able to perform the following functions:

- Add new animals/crops to the approved animal/crop list
- Validate owner requests to include a new animal/crop
- Validate a property that is invalid
- View a list of all owners in the system and a list of all visitors in the system
- Delete any visitor or owner account and all information tied to it
- Delete a visitor's visit history log
- See which admin verified which property (as an additional column in the confirmed properties table)







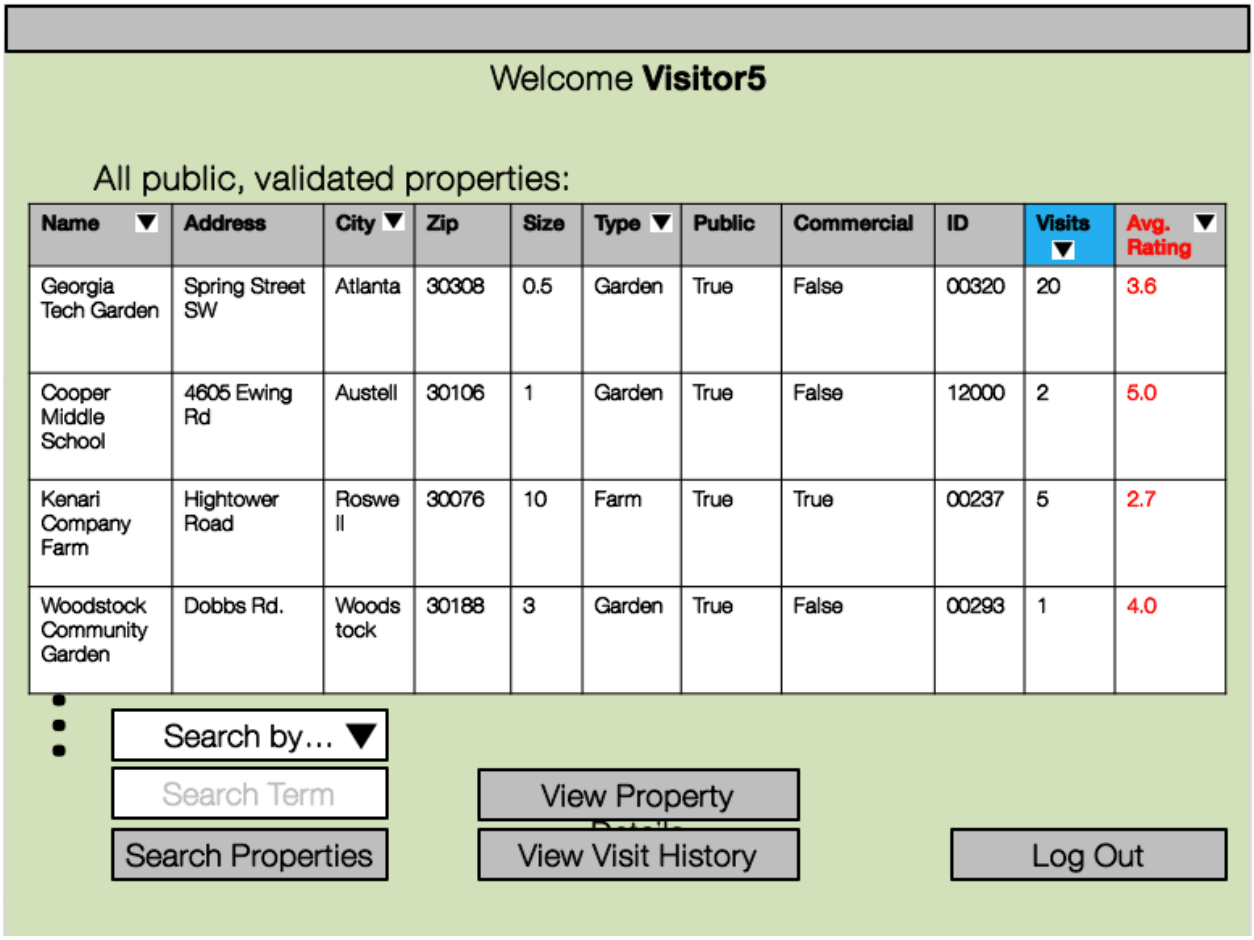


Notes:

- Within the given mockups**: if an admin wants to confirm an unconfirmed property without changing anything else, they must go through the manage property page, and just press save changes**. Upon returning to the main admin landing, the tables should be updated to reflect this change. **You may add additional UI elements that streamline this process if you wish.**

Visitor Functionality

When a visitor logs in, they should be presented with a table containing all **confirmed**, **public** properties.



Visitor Functionalities:

- The table should be sortable/searchable by various columns/attributes
- The visitor should be able to enter a search on the sortable columns (denoted by downward pointing arrows in the mockup on the previous page)
- If searching based on visits or average rating, the visitor should be able to enter a search range (i.e. between 5 and 10 visits, or avg. rating between 3 and 5)
- The visitor should be able to select a particular property to view its details.

- From this detailed view, the visitor can log their visit (or un-log their visit if it was previously logged)
- Visitors can specify that they have visited a property once, and the date that they do this should be logged (not the date that they actually visited)
- A logged visit **requires** that the visitor leaves a rating between 1 and 5
- If an admin or owner modifies a property's attributes, the number of visitors to the property should be **reset to 0**
- Log out of their account
- Visitors should have the ability to view a history of all of the properties that they have visited

Notice: there is a difference in the above screens: the one on the left hasn't logged a visit yet. The one on the right has a visit logged, so the option to un-log is there instead. The rating stars are just a suggestion, you can just use numbers if you want.