Atlanta Zoo

Fall 2018 - CS 4400 Database Project. Version 1

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. We will also provide you with a list of approved technologies for your implementation, and the professor must approve the use of any other alternative technologies. Under 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 Canvas). Slides on database design methodology will be useful for phases 1 and 2: All slides have been posted on Canvas.

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. You must first go to the TA who graded your assignment before going to the Head TA, if the TA who graded your assignment was unable to resolve the issue. This clock starts when the papers are returned to the class, not when you personally get your returned paper.

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 hardcopy form.

Deliverables

Phase 1 - Submit to Canvas and bring 1 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 Canvas 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 your assumptions.
3. Be careful not to include extra or unnecessary relationships. Only things that are stored in the database should be included in the diagram.
4. The logical constraints that you must list cannot be ones that can be specified using ER notation, nor can they be related directly to data types or values.
5. Each team needs to turn in one hard copy (only 1 per team). Every student must upload an electronic copy to Canvas 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 - Submit to Canvas and bring 1 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 Canvas 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 - Submit to Canvas

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).
    a. A set or sequence of SQL statements may be required in order to complete a given task. However, in such cases, the last SQL statement should show the output according to the specification.
    b. Views and nested queries may be used to support the tasks.
3. For the heavy weight project option, your source code for the application.
    a. 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.
    b. Every student must upload an electronic copy to Canvas 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 during the 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% (each) 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.

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 information about animals, exhibits, and shows at a zoo, as well as the staff who work there and those who visit the zoo.

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, visitor, staff, 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 Canvas 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.) My UI doesn't have a home button, but you should add one that takes the user to the starting page for each user.

User Accounts

Login

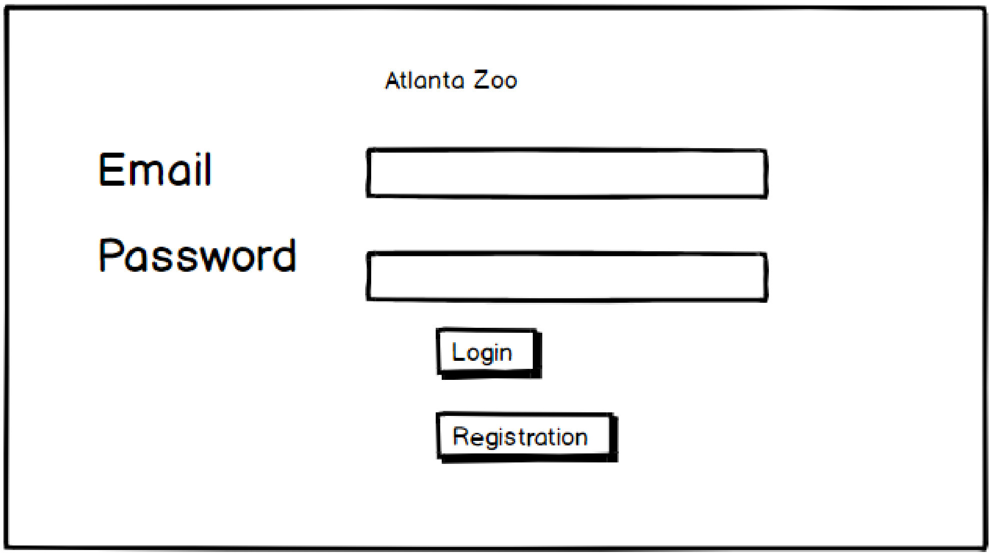
A user must login before using the application. There are three types of users – administrators, staff, and visitors. Each will be described in more detail in their section, but here is a brief overview. Administrators can add animals, shows, and remove visitor or staff accounts. Staff can view their assigned shows that they host and log animal care. Visitors can visit exhibits and shows. To log in, a valid email and password combination is required.

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 (each user only has 1 user type).

• Visitors and staff are the only types of accounts that can be registered (admin accounts will be pre-populated into the database)

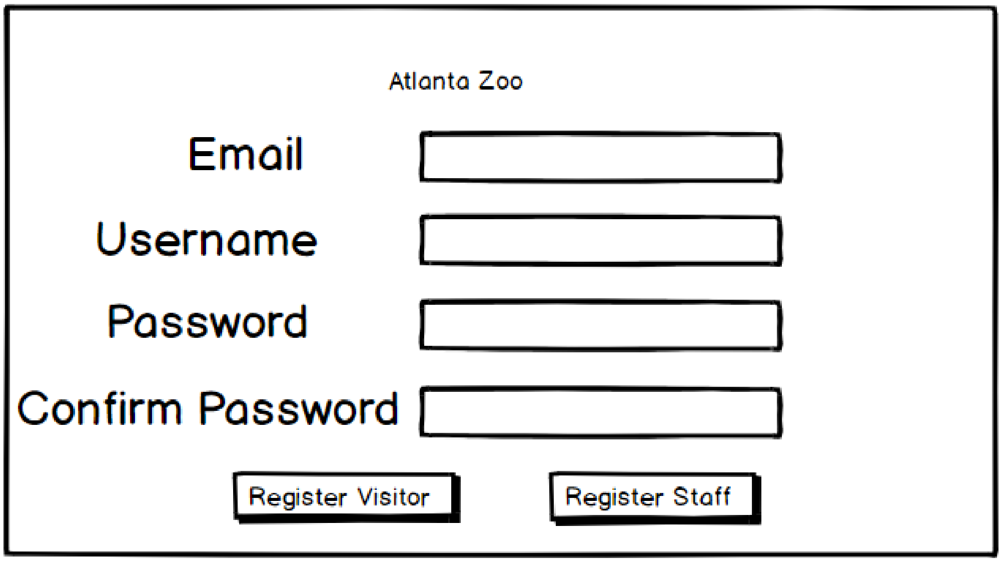


Registration

After clicking registration, the user should be directed to a registration screen. All fields are required. They must specify their user type (staff or visitor).

- Email must be unique
- Username must be unique
- A user can only be a visitor or a staff
- 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 registration page can only be used to create staff and visitor accounts.

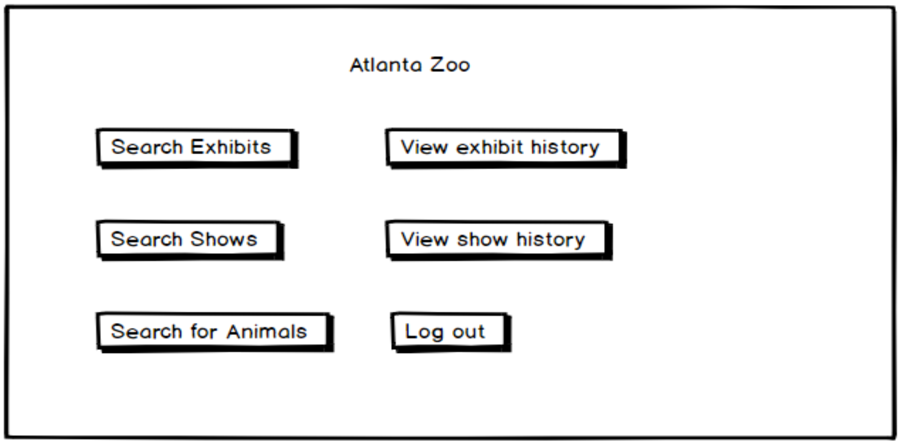
- You may validate the email address using the database (SQL CHECK constraint) or in the front-end application.
- An email address consists of alphanumeric characters, followed by an @ symbol, followed by alphanumeric characters, followed by a . symbol, followed by alphanumeric characters.



Visitor Functionality

When a visitor logs in, they should have a screen with the following options:

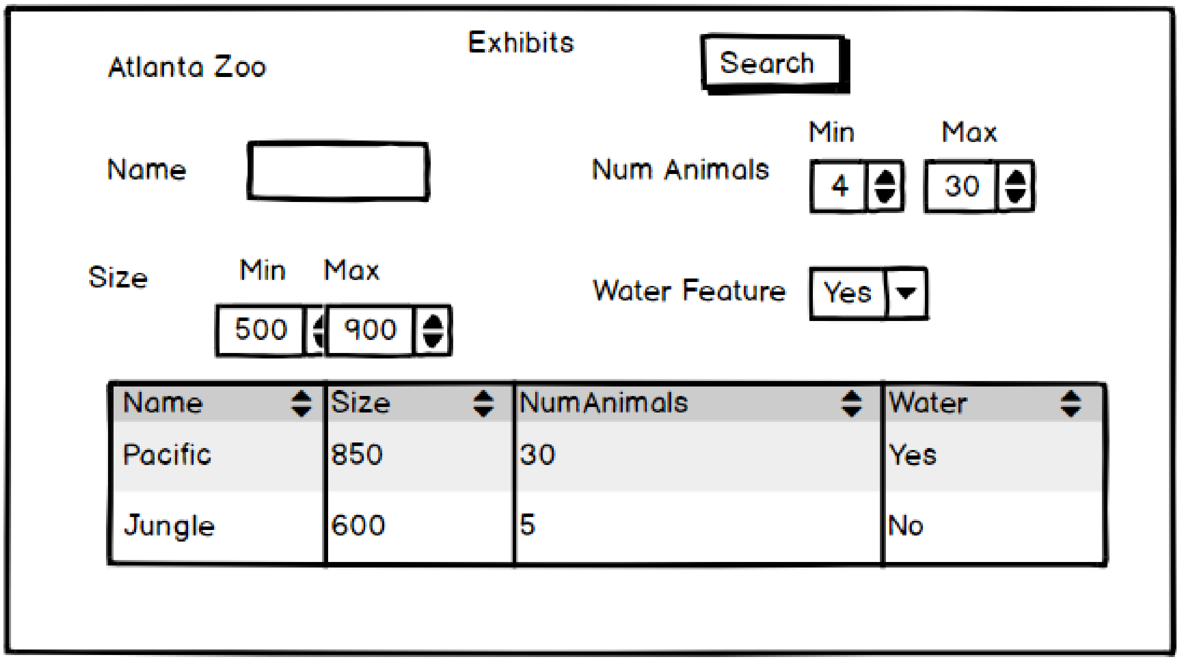
- Search exhibits
- Search shows
- View exhibit history
- View show history
- Search animals
- Log out



Search for Exhibits

A list of the exhibits should be displayed. The user should be able to search the exhibits by any field or combination of fields, and order them by any column. Clicking on an exhibit should go to a detailed view of the exhibit, as shown later on in the project description.
Below are the columns that should be included in this view:

- Size of the exhibit
- Number of animals in the exhibit
- Whether the exhibit has a water feature
- Name of the exhibit. Each exhibit has a unique name.



Name

Num Animals Min 4 Max 30

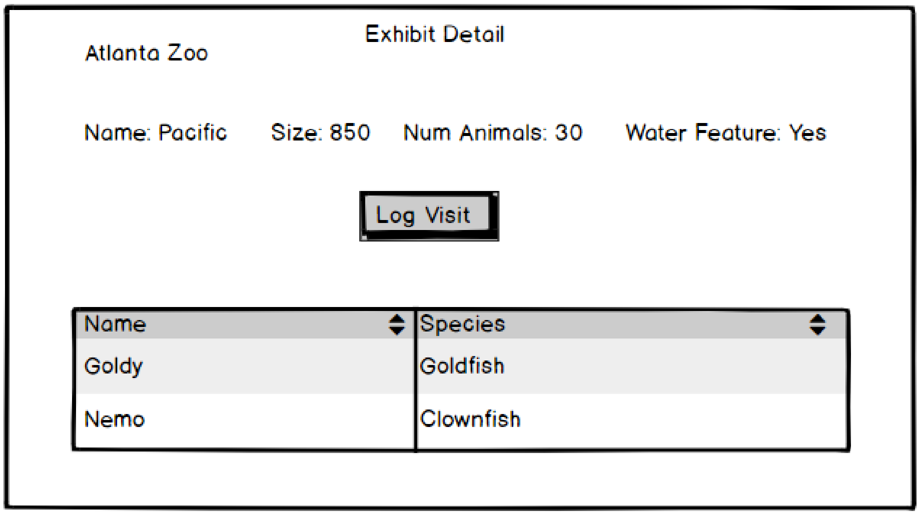
Size Min 500 Max 900

Water Feature Yes

Name	Size	NumAnimals	Water
Pacific	850	30	Yes
Jungle	600	5	No

Exhibit Detail

This view should include the attributes from the Search for Exhibits view, along with a list of the name and species of each animal who lives in the exhibit. If the user clicks on an animal, it should take them to that animals detail page. Visitors can log a visit to an exhibit on this page, which will also log the time of the visit.



Name: Pacific Size: 850 Num Animals: 30 Water Feature: Yes

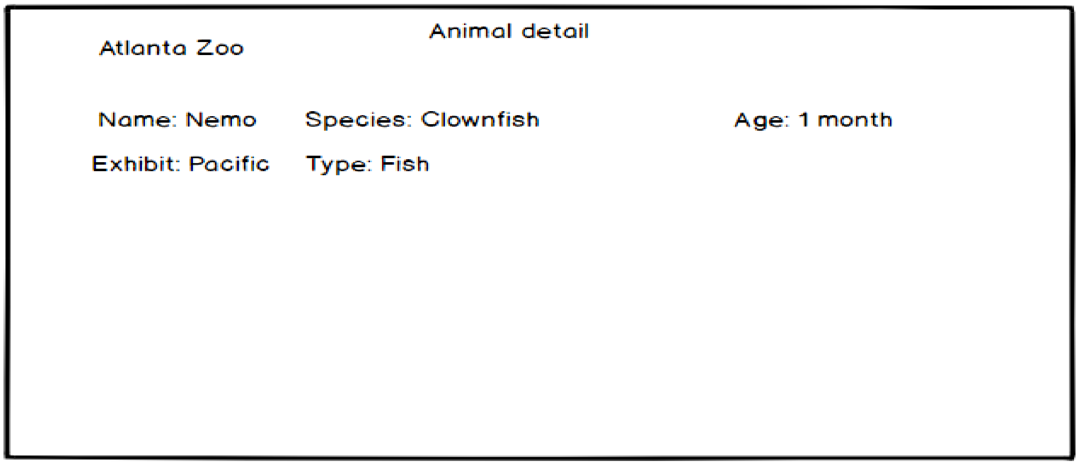
Name	Species
Goldy	Goldfish
Nemo	Clownfish

Animal Detail

This view shows the features of the animal. Here is the information that should be included:

- Animals name
- Animal species
- Type of animal, which can be any of the following values:
  - mammal, bird, amphibian, reptile, fish, or invertebrate
- Age (in number of months)
- Name of the exhibit the animal lives in

The combination of the animals name and the species is unique.



Name: Nemo Species: Clownfish Age: 1 month

Exhibit: Pacific Type: Fish

Search for Animals

The visitors should be able to search for animals by name, type, species, or any combination of those features. The following values should be displayed:

- Name

- Species
- Type
- Age
- Exhibit

Clicking on the exhibit should take the visitor to the exhibit detail page for that exhibit.

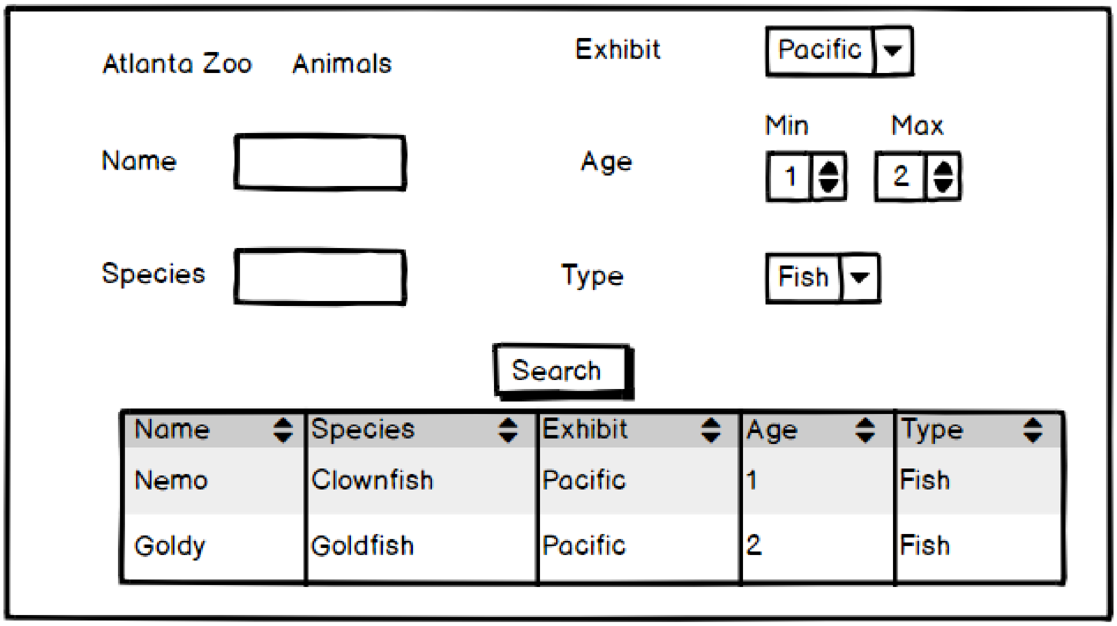


Exhibit Pacific

Age Min 1 Max 2

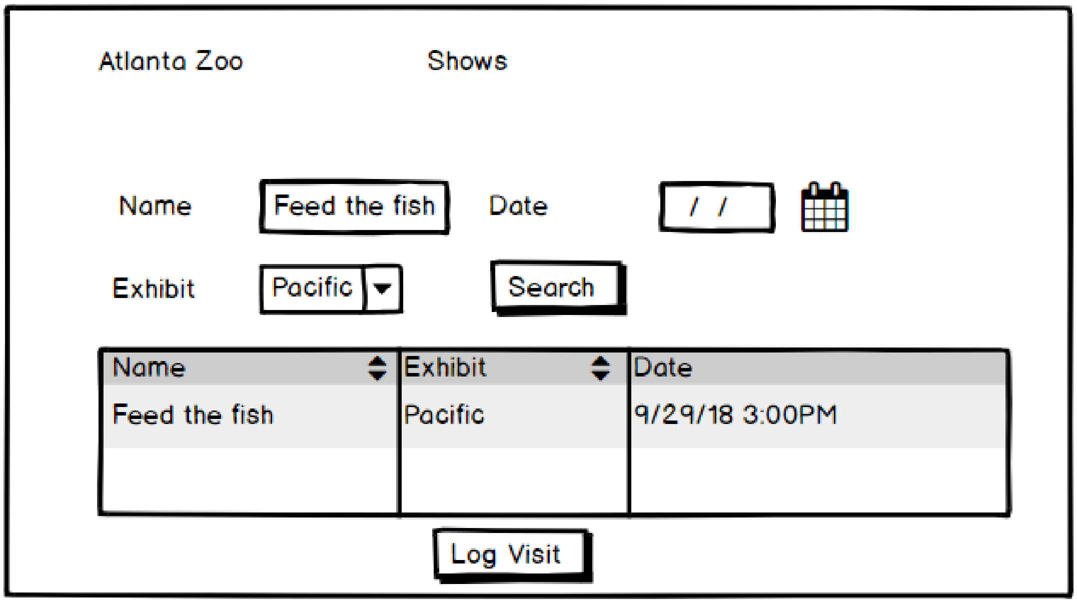
Type Fish

Name	Species	Exhibit	Age	Type
Nemo	Clownfish	Pacific	1	Fish
Goldy	Goldfish	Pacific	2	Fish

Search for Shows

From the visitor home page, the visitor can also search for shows. This view should show all scheduled shows, with the name, time, and exhibit. The visitor should be able to search for shows based off of these features and sort the results. Here, visitors can also log a visit to a show. Visitors should only be allowed to log a visit to a show that has a starting time before or at the current time. Visiting a show should also log a visit to that exhibit for the user. Clicking on an exhibit should take the user to the Exhibit Detail page.

- The time for the show is the time that the show starts.
- The name of the show is not unique.
- See more information about shows in the staff Shows section, and the administrator Add Show section.



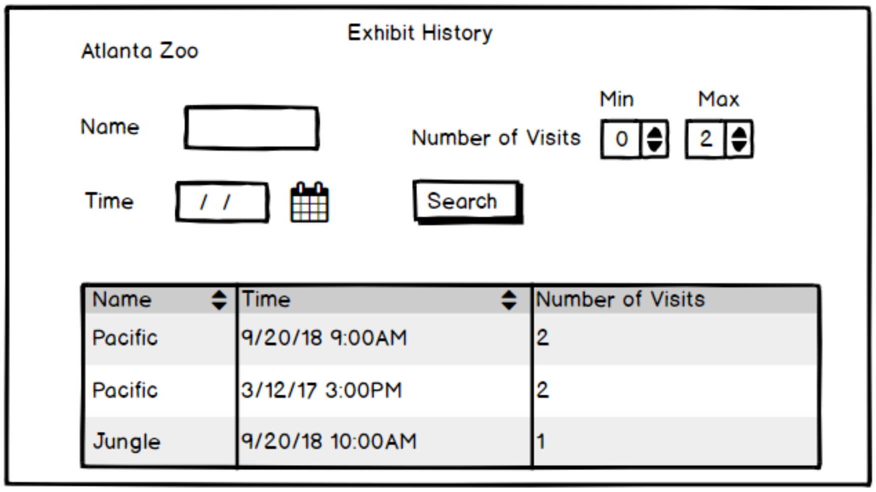
Name: Feed the fish

Exhibit: Pacific

Name	Exhibit	Date
Feed the fish	Pacific	9/29/18 3:00PM

Exhibit History

This view should show the visitor all of the exhibits to which they have logged visits. It should show the time of the visit and the name of the exhibit, along with the total number of visits to the exhibit. Visitors should be able to search their history by exhibit name, time, or total number of visits. You can see here that exhibits can be visited multiple times by the same user, and the number of visits is the number of visits that user has made to that exhibit. Clicking on an exhibit here should go to the exhibit detail page.



Number of Visits: Min 0, Max 2

Name	Time	Number of Visits
Pacific	9/20/18 9:00AM	2
Pacific	3/12/17 3:00PM	2
Jungle	9/20/18 10:00AM	1

Show History

This view should show the visitor all of the shows they have logged visits to. They should see the name, time, and exhibit for the show and be able to search for shows.

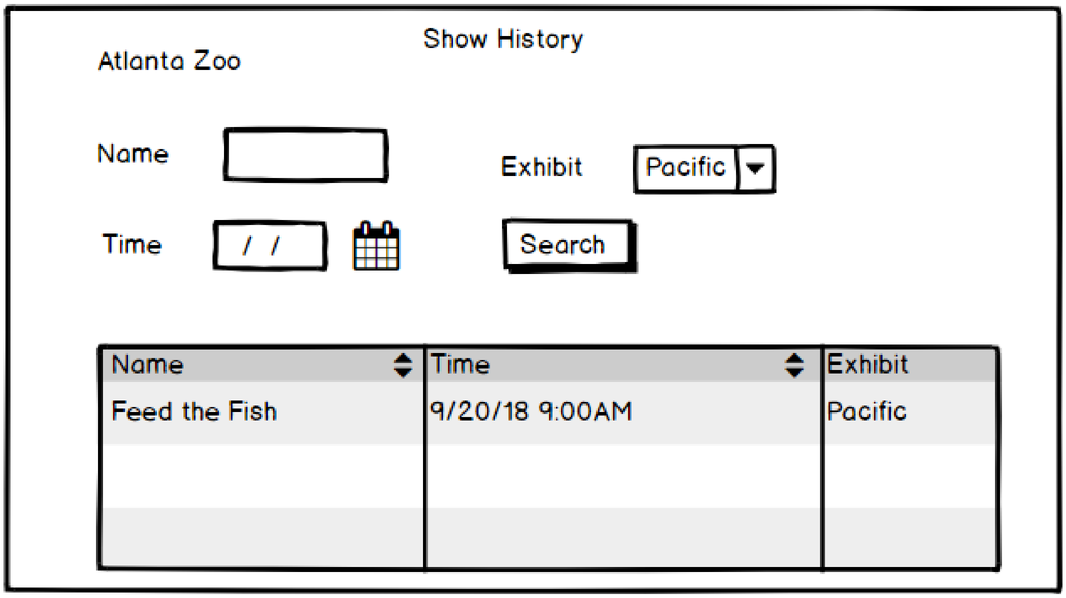


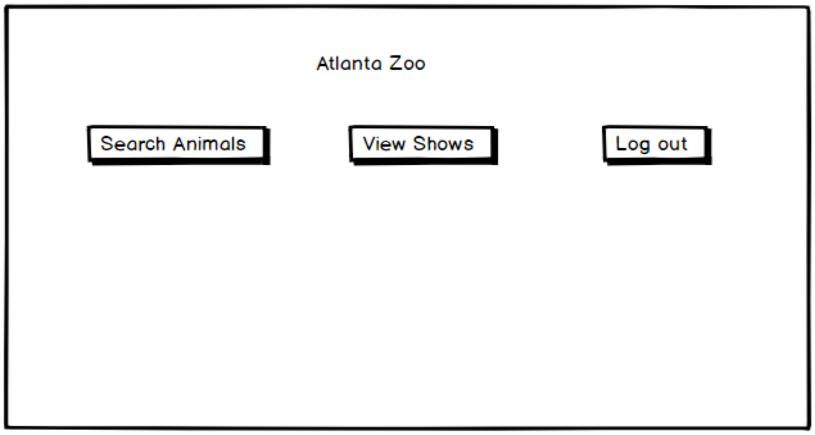
Exhibit: Pacific

Name	Time	Exhibit
Feed the Fish	9/20/18 9:00AM	Pacific

Staff Functionality

When a staff member logs in, they should have the following options:

- View assigned shows
- Search Animals.
- Log out

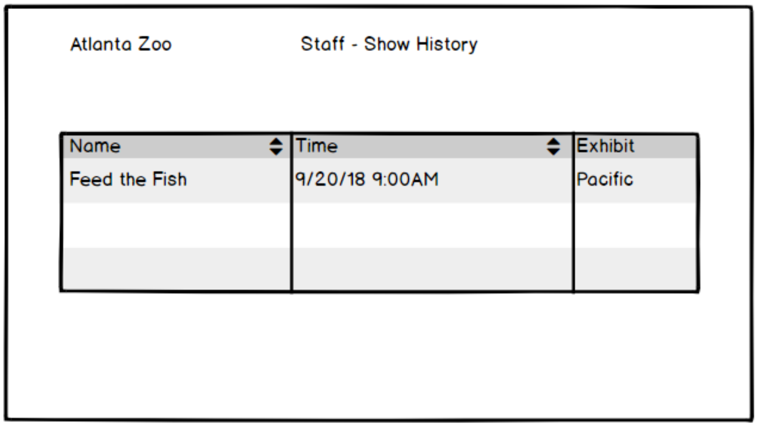


Shows

The staff member should see all shows for which an admin has assigned them to host, regardless of if the show has already occurred or not. They should not see shows for which they are not the host. The following attributes should be displayed:

- Time of show
- Name of show
- Exhibit the show is located in

Please reference the “Add show” section in the administrator functionality for more information about shows.

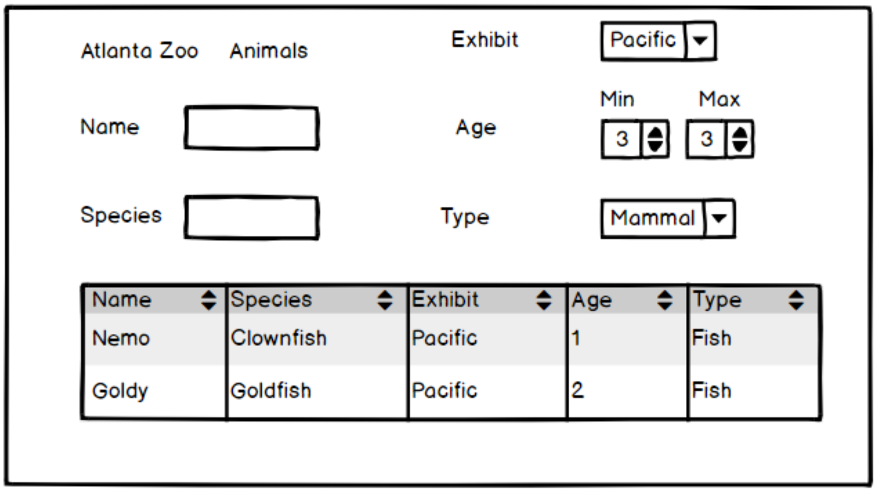


Atlanta Zoo   Staff - Show History

Name	Time	Exhibit
Feed the Fish	9/20/18 9:00AM	Pacific

Animals

Here staff can search for animals by name, species, or any other attribute or combination of attributes. A staff member can select an animal, which takes them to the animal care page where they can log the care that the animal received from the staff and see the animals history of care.



Atlanta Zoo   Animals

Exhibit: Pacific

Name:

Age: Min 3 Max 3

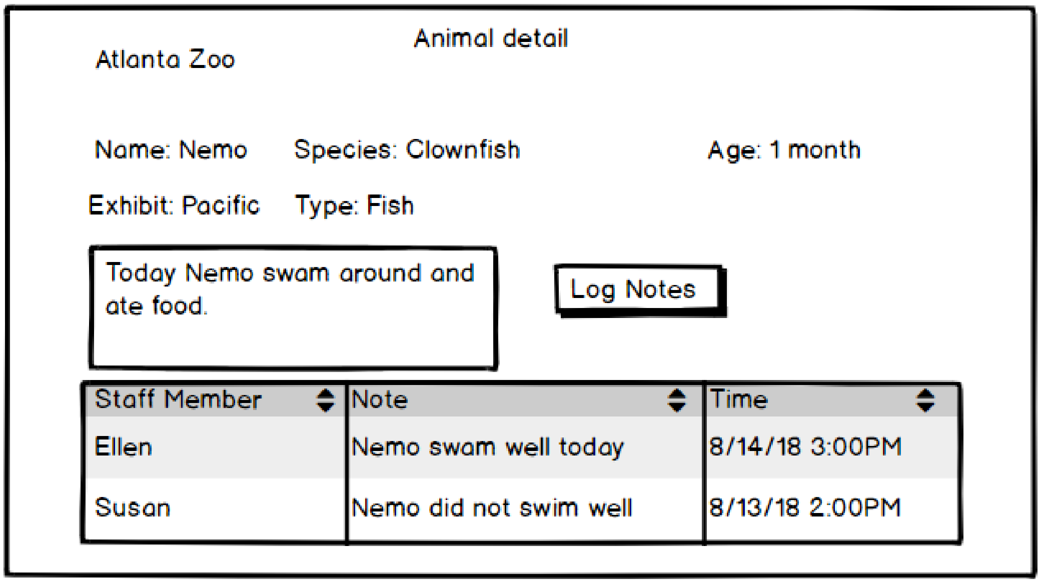
Species:

Type: Mammal

Name	Species	Exhibit	Age	Type
Nemo	Clownfish	Pacific	1	Fish
Goldy	Goldfish	Pacific	2	Fish

Animal Care

Staff can write a note about the animal, and record it here so other staff members can view it later, along with the time of the note. Note that we also keep track of which staff member logged the information. See the UI below.



Animal detail

Atlanta Zoo

Name: Nemo Species: Clownfish Age: 1 month

Exhibit: Pacific Type: Fish

Today Nemo swam around and ate food.

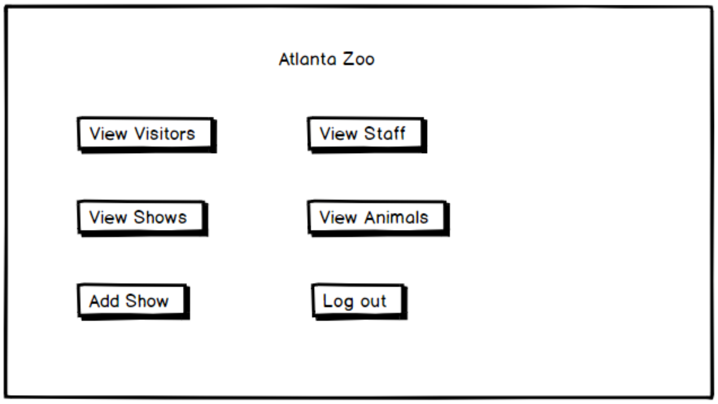
Staff Member	Note	Time
Ellen	Nemo swam well today	8/14/18 3:00PM
Susan	Nemo did not swim well	8/13/18 2:00PM

Administrator Functionality

When an administrator logs in, they should have the following options:

- View visitors
- View staff
- View shows
- View Animals
- Add Animals
- Add show

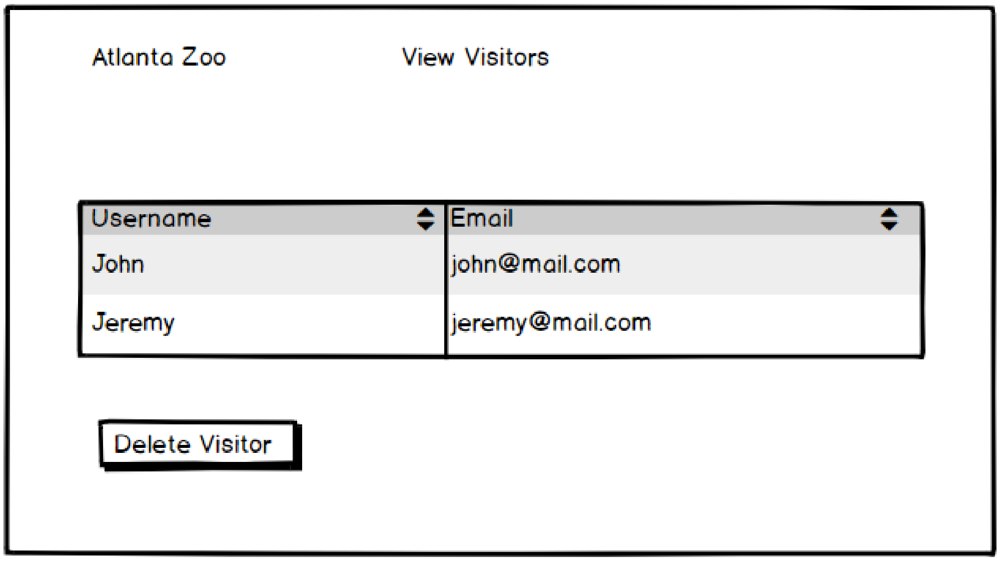
Exhibits cannot be added or deleted.



Atlanta Zoo

View Visitors

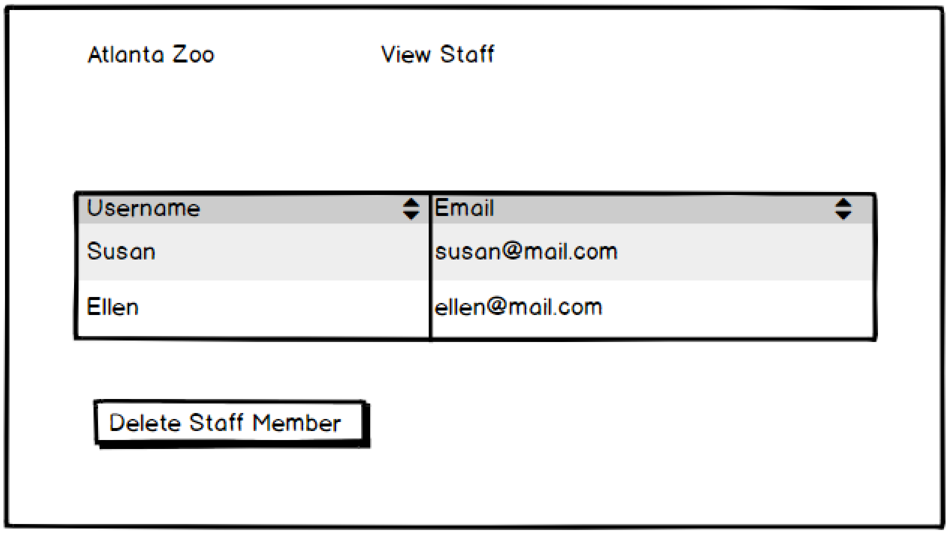
The admin should see a list of all the visitors with their username and email. The admin should be able to search the list of visitors. The admin can then remove visitor accounts, which would delete all information about the visitor.



Username	Email
John	john@mail.com
Jeremy	jeremy@mail.com

View Staff

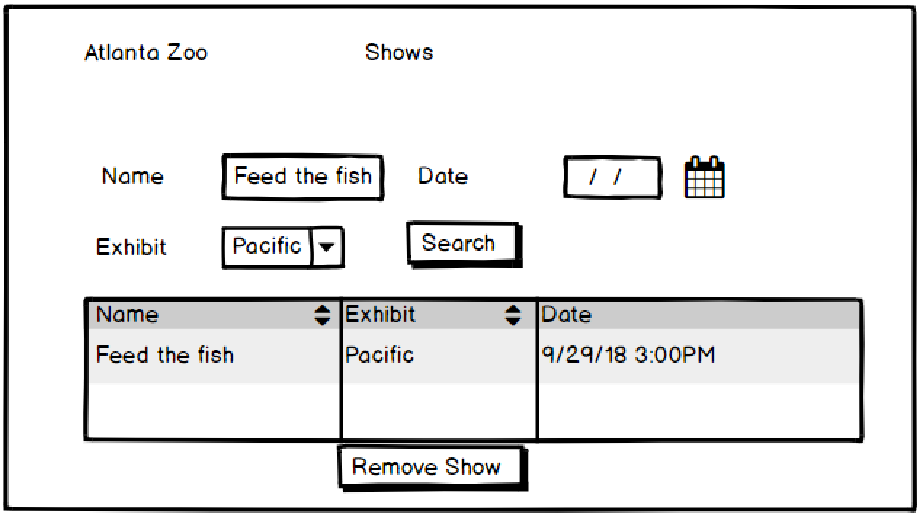
The admin should see a list of all the staff members with their username and email, which the admin can search. The admin can remove staff accounts, which also removes all of the information about the staff member.



Username	Email
Susan	susan@mail.com
Ellen	ellen@mail.com

View Shows

The admin can see the list of shows, and can search for shows. The admin can also remove shows. If a visitor has logged a visit to a show that was removed, that visit is also removed.



Name: Feed the fish

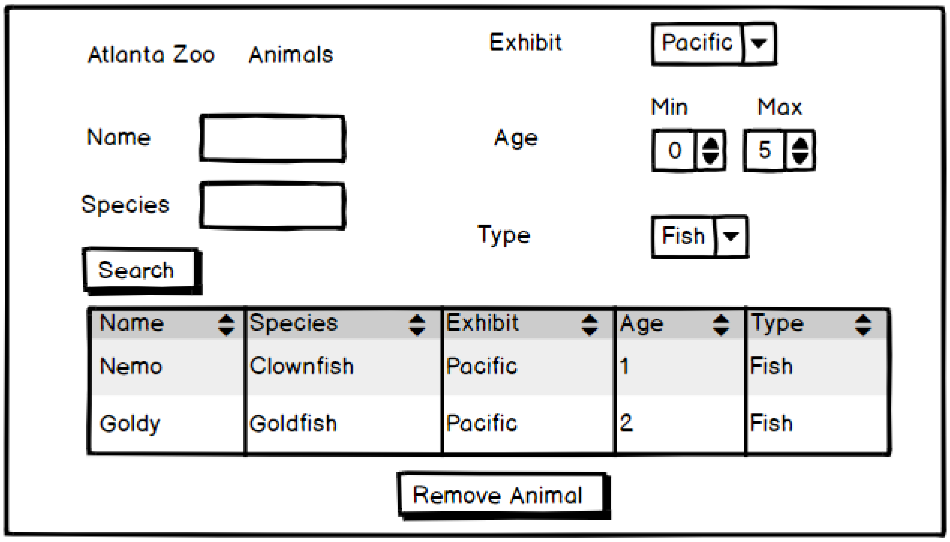
Date: / /

Exhibit: Pacific

Name	Exhibit	Date
Feed the fish	Pacific	9/29/18 3:00PM

View Animals

The admin can view all the animals and search for animals. Animals can be removed by the admin.



Name:

Species:

Exhibit: Pacific

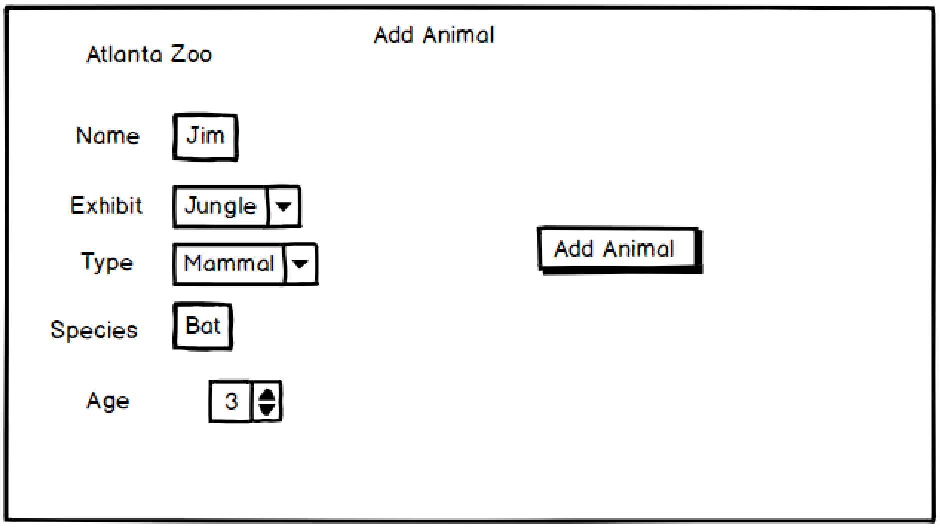
Age: Min 0 Max 5

Type: Fish

Name	Species	Exhibit	Age	Type
Nemo	Clownfish	Pacific	1	Fish
Goldy	Goldfish	Pacific	2	Fish

Add Animals

The admin can add a new animal, and must supply all fields for the animal. These include name, species, type, age, and exhibit.


Atlanta Zoo

Add Animal

Name: Jim

Exhibit: Jungle

Type: Mammal

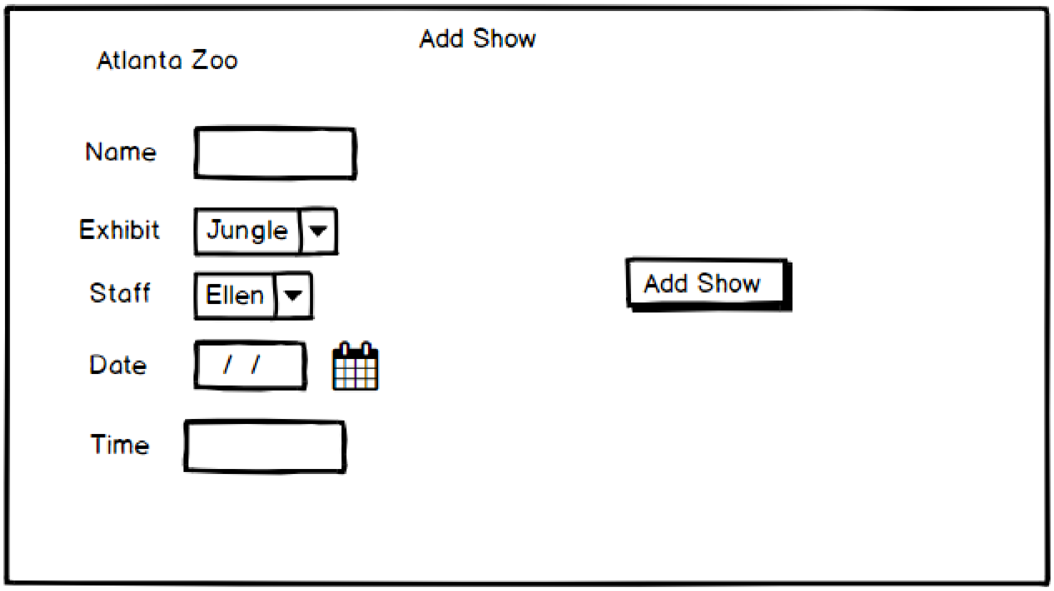
Species: Bat

Age: 3

Add Shows

The admin can add a new show to the schedule. The admin must provide a staff member to host the show, a name for the show, an exhibit for the show, and a time for the show. Here are a few more notes about shows.

- A staff member cannot host multiple shows at the same time.
- Each show needs one and only one host.
- It is possible for multiple shows to occur at the same time in an exhibit.


Atlanta Zoo

Add Show

Name:

Exhibit: Jungle

Staff: Ellen

Date: / /

Time: