Revision History

Version	Date	Comment
1.0	2/7/2019	Initial Release

Atlanta BeltLine

Spring 2019 - 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. You will use your localhost MySQL Server (Version 5.1 or above) 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 three sections (A & B & C) 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

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)
    a. You MUST include the screen numbers (see figure caption and figure header) when making IDF. Failure to do so will result in point deduction
    b. An example IFD is provided, you don't have to include the screens from the example in your 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. Also, information extraneous to the required functionality may count against you
3. 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
    a. (i.e. "password must have at least 8 characters," "transport type include 'MARTA,' 'Bus'" are NOT logical constraints)
4. 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

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.

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 that stores information about the Atlanta BeltLine sites and events. The intent of the application is to keep track of the visitors who visit the different sites and events, users who take transits that connect various sites, and the employees who work in the Atlanta BeltLine system.

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: 1. Tables – they have arrows to indicate which columns/attributes are sortable; 2. Dropdown lists, checkboxes, and radio; 3. Multiselect list that allow multiple selections (see Screen 21 Administrator Edit Transit).

Filtering should be an option to reduce the size of any table. Depending on the table, the filter criteria will be different (different tables have different columns). The description is grouped in the following sections: General Functionalities, User/Employee Functionality, Administrator Functionality, Manager Functionality, Staff Functionality, Visitor Functionality.

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.

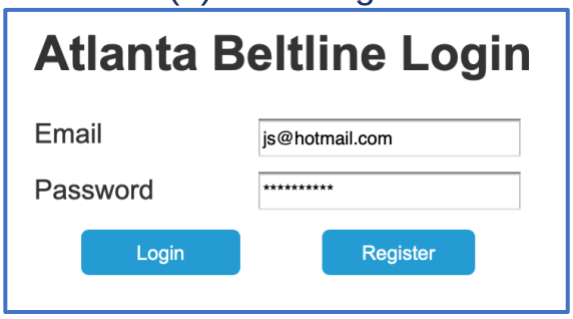
For the Heavyweight option, you may implement the project as a traditional standalone application (e.g. using Java GUIs) or as a web application (e.g. using a web scripting language like PHP). There is a list on Canvas about which languages/tools/software/platforms are allowed. A Piazza post will be 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.)

Notes regarding all applicable screens:

1. Dropdown lists (See Screens)
    a. All information in a dropdown list must be extracted from the database (DO NOT hardcode) unless specified otherwise
    b. All dropdown list must contain an additional “ALL” or “Other” option, this option can be hardcoded
2. Filter criteria – text field
    a. If a text field is left blank during filtering, you can discard this filter criterion (i.e. this filter does not restrict result entries)
    b. All text fields for filter can be left blank unless specified otherwise
    c. If a text field is filled, you must include this filter criterion and perform a partial match (DO NOT do an exact match) based on the information entered unless specified otherwise
3. Filter criteria – date
    a. If a text field is also a date field (*Start Date, End Date*), you must perform an exact match based on the information entered unless specified otherwise
    b. The filtered results should be inclusive
4. Filter criteria – range
    a. The range contains a lower bound and an upper bound.
    b. If the lower bound is left blank, it is assumed to be 0
    c. If the upper bound is left blank, you can assume it to be a very large number
    d. If filled, the results should contain entries are within the range inclusive
5. Tables
    a. The default display (before the first click of “Filter” button) is an empty table
    b. Filter results are ordered by the first sortable table column after filtering
    c. All table columns with a pair of upward and downward triangles are required to be sortable; you must use SQL commands for sorting (DO NOT do sorting in the frontend)
6. Button click
    a. For heavyweight, you must have a reasonable design of button events (i.e. which screen should the UI be directed to after clicking each button, including the “Back” button)
    b. For heavyweight, when performing an action, you must prompt meaningful messages (e.g. prompt a “password and confirm password does not match” whenever necessary)

General Functionalities

(1) User Login

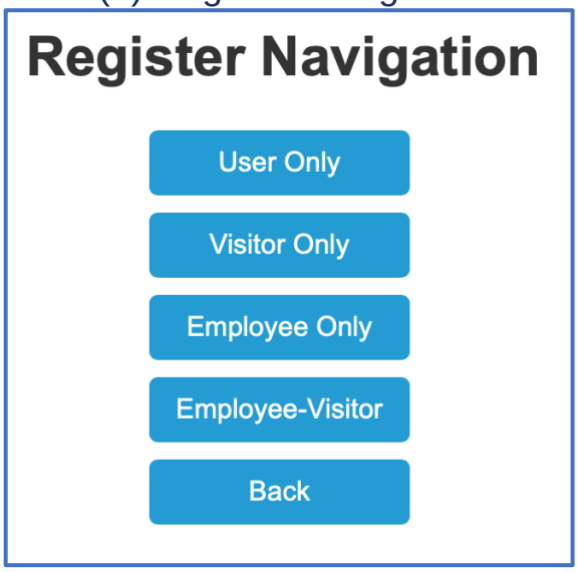


Screen 1 User Login

Notes:

1. All user must log in before using the application
2. All user must be directed to the correct functionality screens after successful login
3. All users share the same login screen
4. Email and password combination must exist in the database in order for a successful login
5. User accounts must be approved by administrators for login
6. User cannot login with declined accounts, nor can they register for another account using the same information
7. A user can be a visitor, an employee, both, or just a simple user. An employee has to be one of the following: administrator, manager, staff
8. New users (except for admin) must register before login

(2) Register Navigation



Screen 2 User Register Navigation

Notes:

1. There are four options for registering accounts (see Screen 2 User Registration)

(3) Register User Only

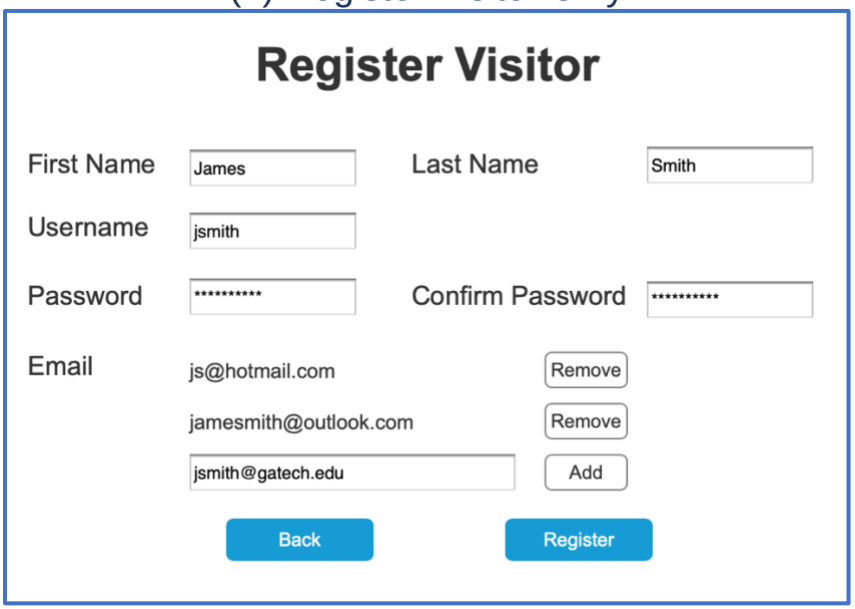
Screen 3 Register User Only

Notes:

1. All fields are required

2. *Username* is unique amongst all users
3. *Password* and *Confirm Password* must match; *Password* must have at least 8 characters
4. You must perform a hash on password before storing it in the database
5. *Email* has the following format: alphanumeric values + "@" + alphanumeric values + "." + alphanumeric values
6. *Email* is unique amongst all users; a user can add more than one unique email to their account
7. All newly registered user accounts will be "pending" for administrator's approval
8. If *Create Visitor Account* is selected, the user is then registered as a visitor
9. Account type will be "User"

(4) Register Visitor Only

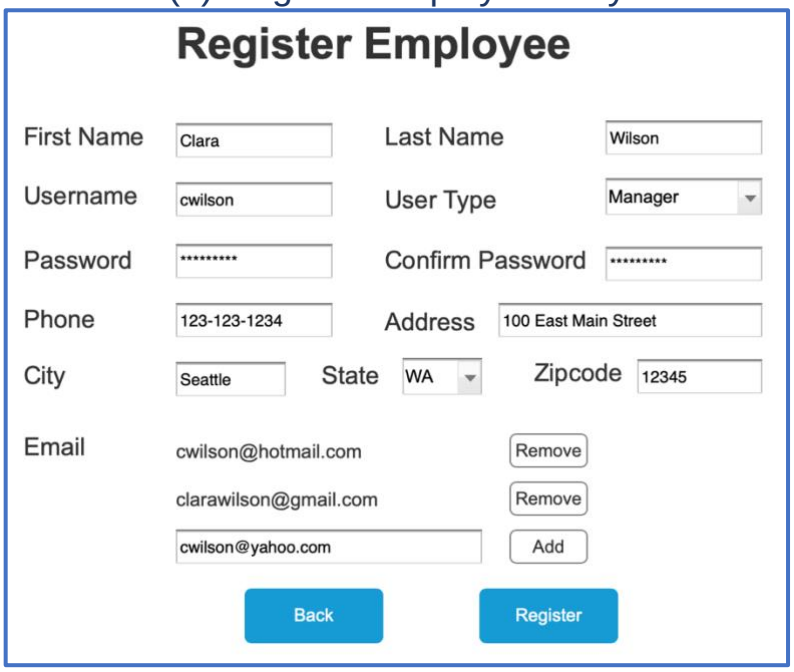


Screen 4 Register Visitor Only

Notes:

1. Same requirements as Screen 3 Register User Only
2. Account type will be "Visitor"

(5) Register Employee Only

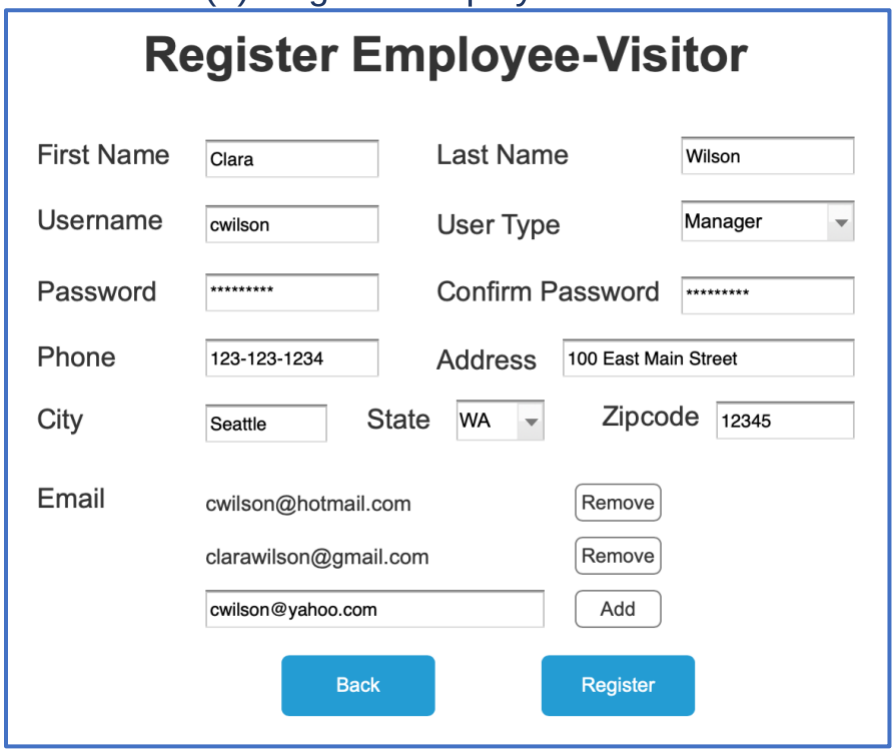


Screen 5 Register Employee Only

Notes:

1. Similar to Screen 3 Register User Only
2. *State* is a dropdown list containing the abbreviations of 50 States in US and "Other;" you can hardcode this dropdown list
3. *User Type* is a dropdown list containing "Manager" and "Staff;" you can hardcode this list
4. Administrators cannot be registered, and they are prepopulated accounts in the database
5. *Phone* is a 9-digit number
6. *Zipcode* is a 5-digit number
7. If *Create Visitor Account* is selected, the employee can also be a visitor; however, their user type is not "Visitor"
8. *Phone* is unique among all employees
9. Account type will be "Employee"

(6) Register Employee-Visitor

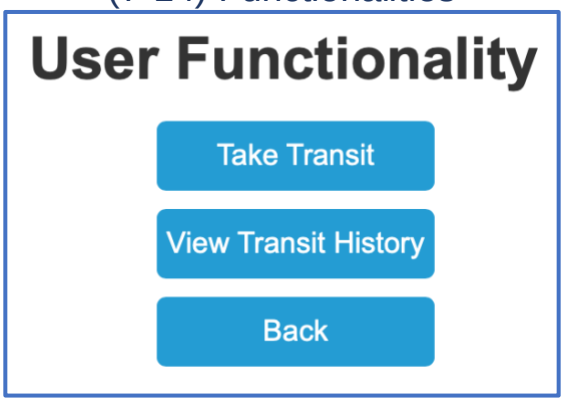


Screen 6 Register Employee-Visitor

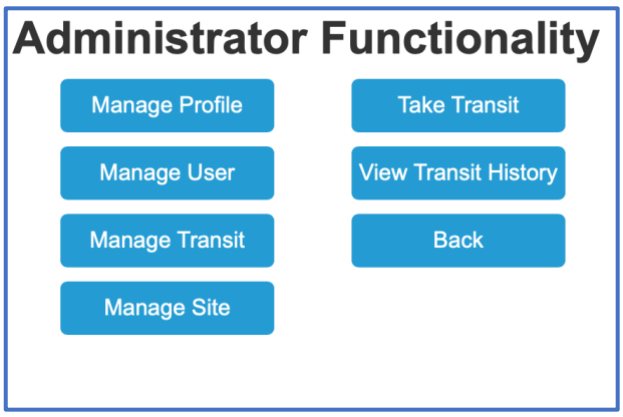
Notes:

1. Same requirements as Screen 5 Register Employee Only
2. Account type will be “Employee”

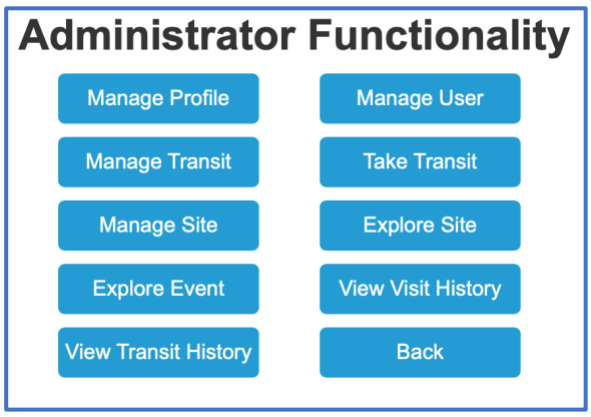
(7-14) Functionalities



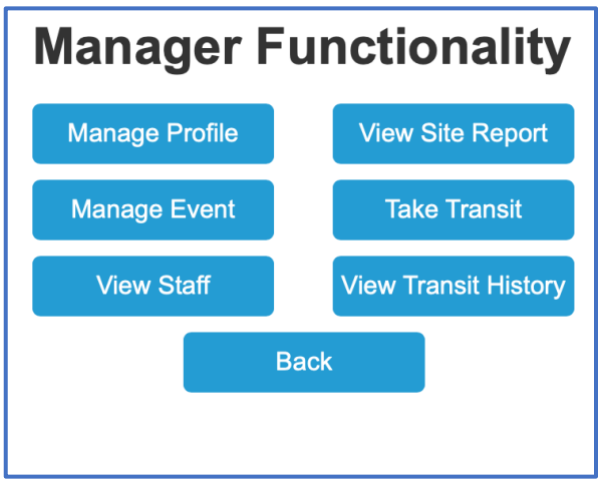
Screen 7 User Functionality



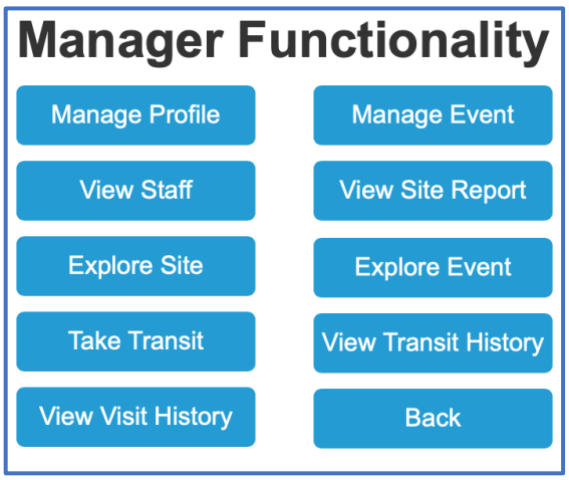
Screen 8 Administrator-Only Functionality



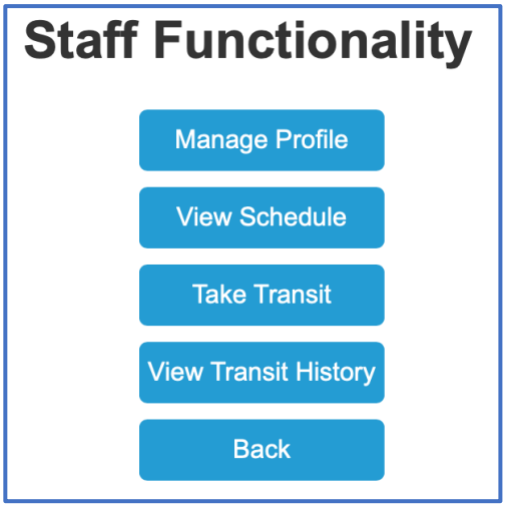
Screen 9 Administrator-Visitor Functionality



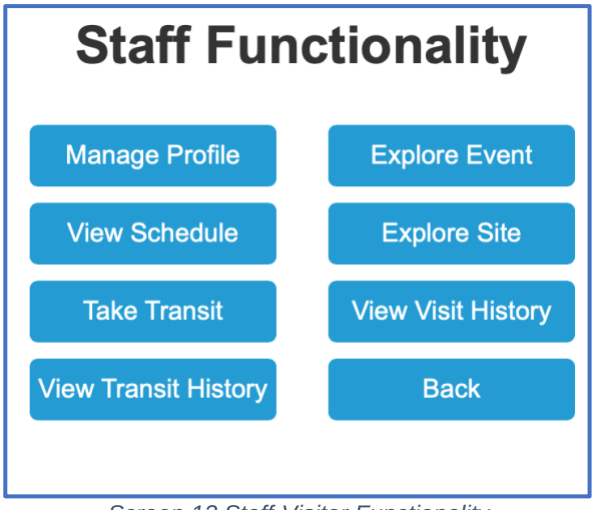
Screen 10 Manager-Only Functionality



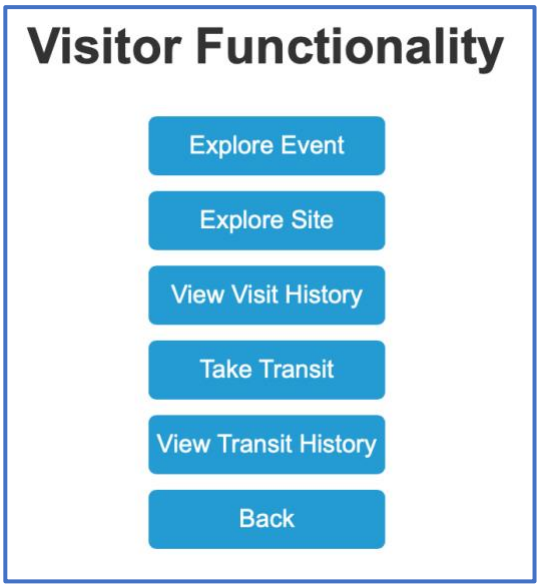
Screen 11 Manager-Visitor Functionality



Screen 12 Staff-Only Functionality



Screen 13 Staff-Visitor Functionality



Screen 14 Visitor Functionality

Notes regarding functionalities:

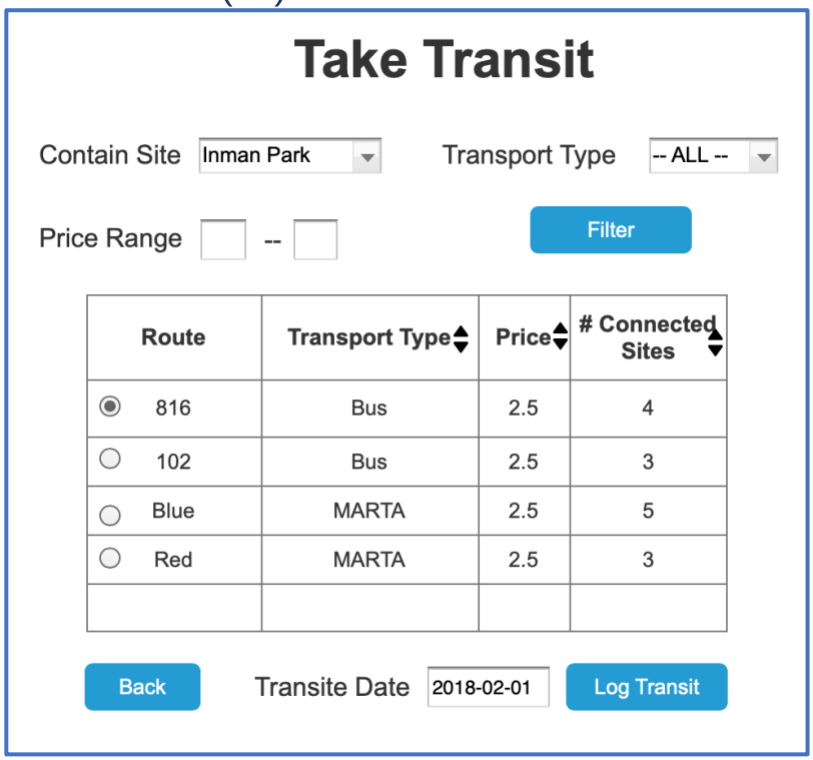
1. User
    a. All users can take existing transits between sites
    b. All users can view their transit history
    c. User can be visitor, employee, or just a user
2. Employees
    a. An employee is a user
    b. All employees can manage their profile
    c. Employee can take one of the three roles: Administrator, Manager, and Staff
3. Administrator
    a. An administrator is an employee
    b. Administrator accounts are prepopulated in the database
    c. Administrator can approve or decline registered accounts
    d. Administrator can create new sites as well as delete/edit existing sites
    e. Administrator can create new transit route and delete/edit existing transit routes
4. Manager
    a. A Manager is an employee
    b. Each site must be managed by one and only one manager
    c. Each manager can at most manage one site
    d. Manager can create new event and edit/delete existing event for their site
    e. Manager can assign available staffs to the events on their site
    f. Manager can view staffs for all sites
    g. Manager can view the report for their own site
5. Staff
    a. A Staff is an employee

b. Staff can view their own assigned schedules
c. Staffs can be assigned to multiple events across multiple sites if they are available for the event dates

6. Visitor
   a. A Visitor is a user
   b. Visitor can explore different sites and log their visits to these sites
   c. Visitor can explore different events and log their visits to the events with remaining tickets
   d. Visitor can visit sites and events for multiple times if possible
   e. Visitor can view their visit history
   f. Visiting a site is free, visiting an event might not be free

User/Employee Functionality

(15) User Take Transit

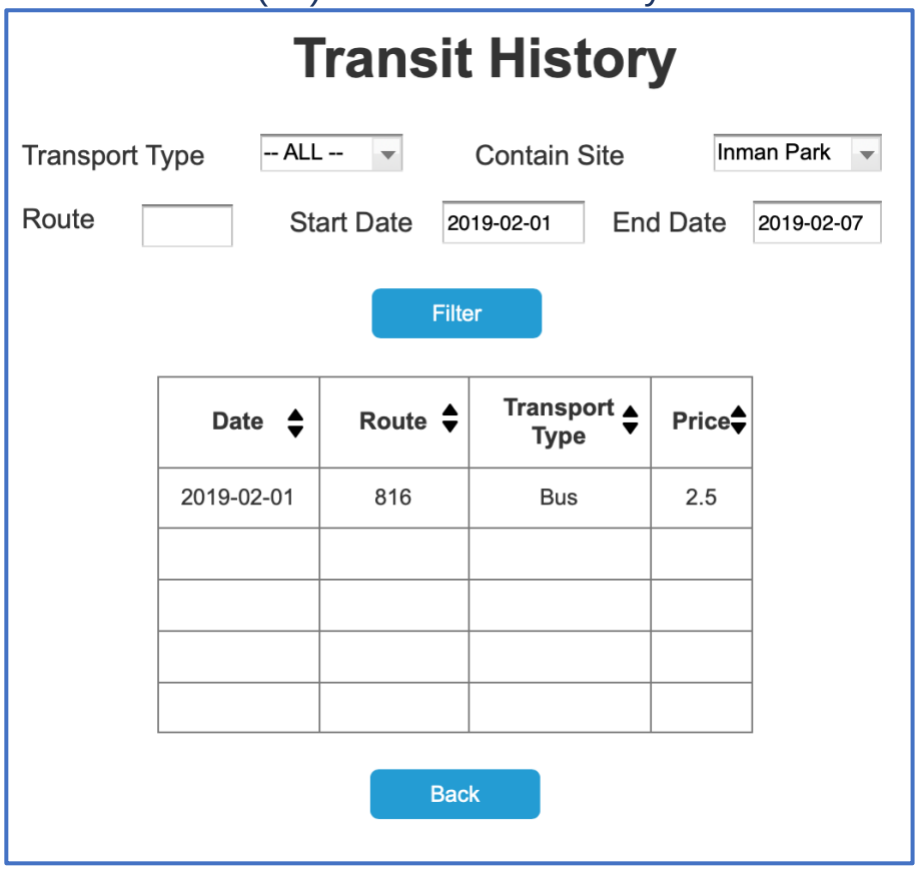


Screen 15 User Take Transit

Notes:

1. A *Transit* connects multiple sites, and a site can be connected via multiple *Transit*
2. Must select a *Route* and enter a *Transit Date* before *Taking Transit*
3. *Transport Type* is a dropdown list containing “MARTA,” “Bus,” and “Bike;” you can hardcode this dropdown list
4. User can only take the same transit once a day

(16) User Transit History

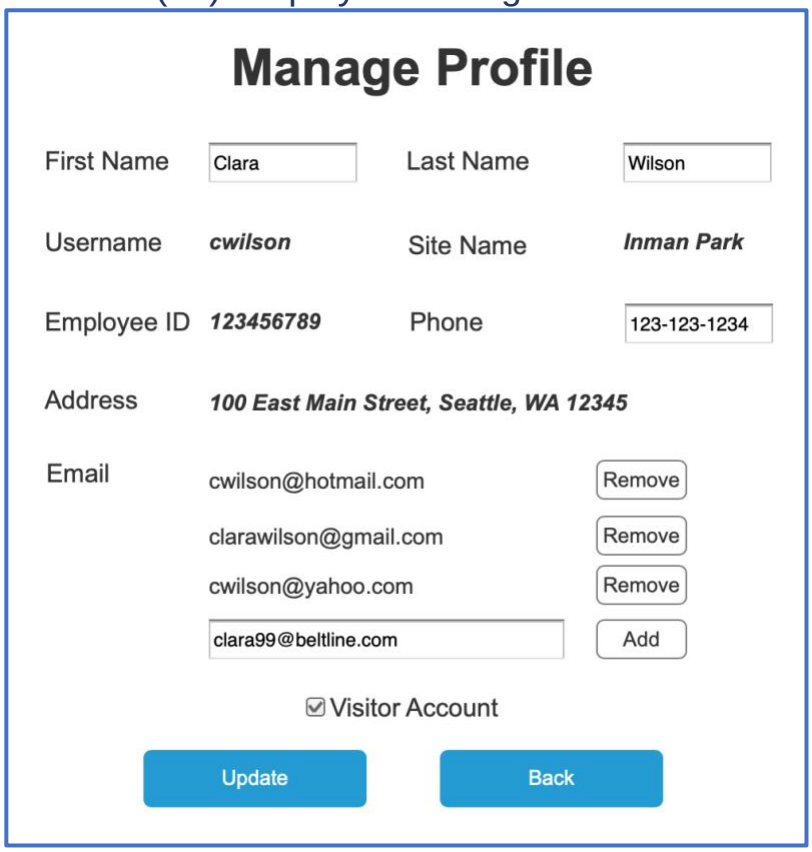


Screen 16 User Transit History

Notes:

1. *Transport Type* see Screen 15 User Take Transit Note 2
2. *Route* filter criterion must be matched exactly if filled in

(17) Employee Manage Profile



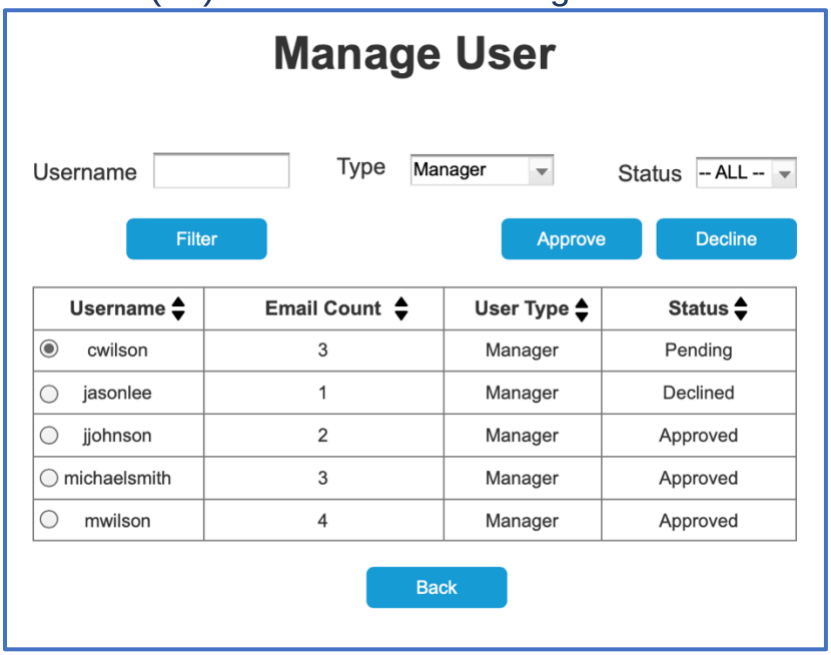
Screen 17 Employee Manage Profile

Notes:

1. *First Name, Last Name, Phone, Email, Visitor Account* is editable
2. *Visitor Account* is checked if the employee is also a visitor
3. Deselecting *Visitor Account* and *Update* will erase all visit history for the employee; The employee will lose access to all visitor functionalities
4. *Employee ID* is a unique 9-digit number assigned to each employee once their account got approved by administrator (you can auto-generate this number)

Administrator Functionality

(18) Administrator Manage User



Screen 18 Administrator Manage User

Notes:

1. *Status* is either “Approved,” “Pending,” or “Declined;” you can hardcode this dropdown list
2. *Type* contains “User,” “Visitor,” “Staff,” and “Manager;” you can hardcode this dropdown list
3. *Email Count* is the number of emails the user registered
4. *Username* filter criterion must be matched exactly if filled
5. Administrator can approve a pending (or declined) account, can decline a pending account, but cannot decline an approved account

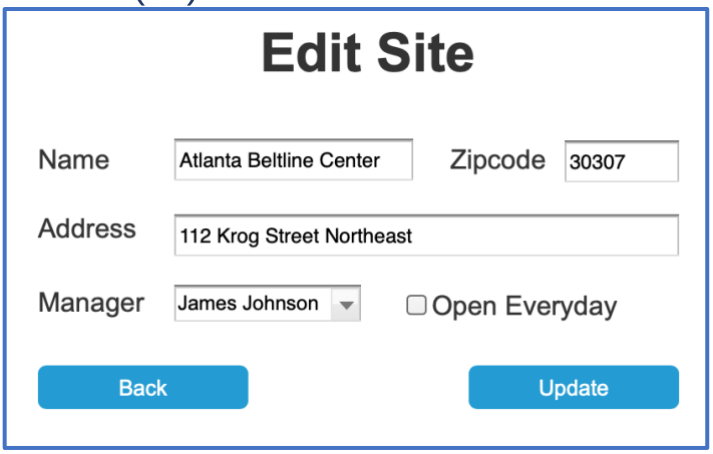
(19) Administrator Manage Site

Screen 19 Administrator Manage Site

Notes:

1. *Open Everyday* is a dropdown list containing "Yes," and "No;" you can hardcode this list
2. Administrator must select one existing site before editing or deleting it
3. Deleting a site will delete all the corresponding events in the site, staff assignments for the events, visits logged to the events, visits logged to the site, transits related to the site, and transit history related to the site
4. Editing a site might change the site information for all the corresponding events in the site, visits logged to the events, visits logged to the site, transits related to the site, and transit history related to the site

(20) Administrator Edit Site

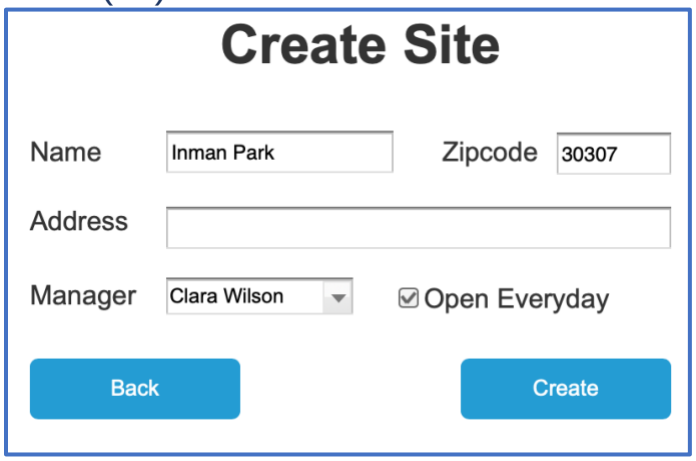


Screen 20 Administrator Edit Site

Notes:

1. *Manager* must be a dropdown list containing the current site manager as well as the managers who have not yet been assigned to another site
2. All fields are editable

(21) Administrator Create Site

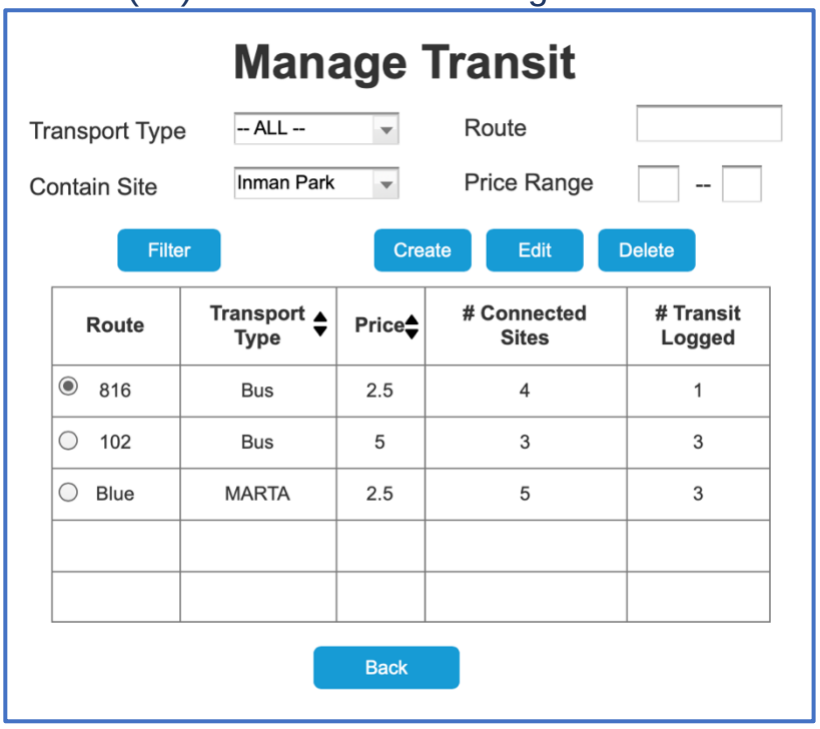


Screen 21 Administrator Create Site

Notes:

1. All fields except for *Address* are required
2. *Zipcode* is a 5-digit number
3. *Name* must be unique for all sites
4. *Manager* must be a dropdown list containing ONLY the managers who have not yet been assigned to a site
5. Site has no entry fee

(22) Administrator Manage Transit

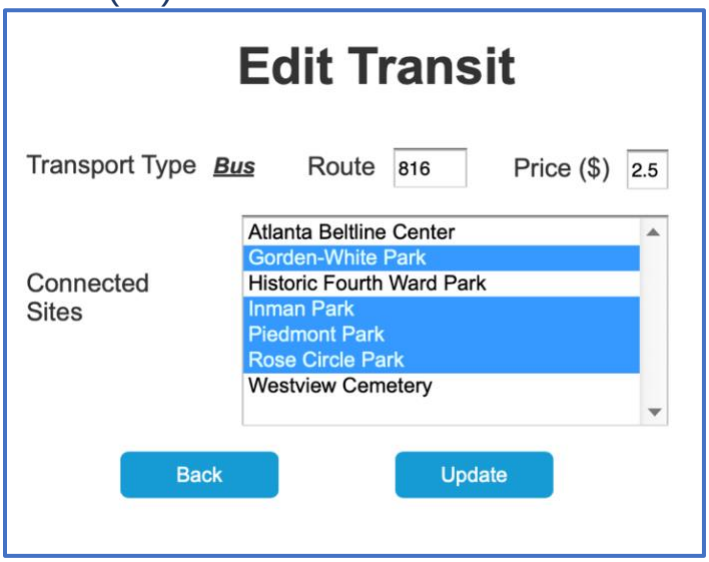


Screen 22 Administrator Manage Transit

Notes:

1. Administrator must select an existing transit before editing or deleting it
2. Deleting a transit will delete all the corresponding transit history
3. Editing a transit might change all the corresponding transit history
4. *Route* filter criterion must be matched exactly if filled
5. *Transport Type* see Screen 15 User Take Transit Note 2

(23) Administrator Edit Transit

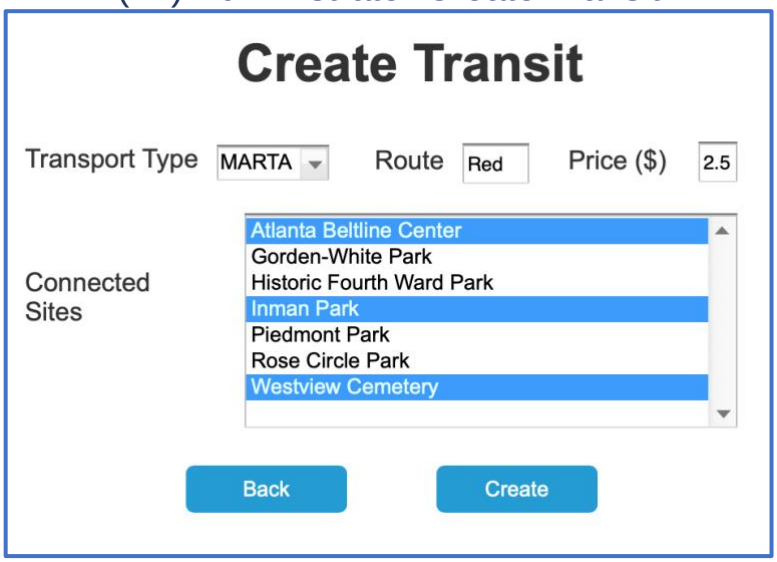


Screen 23 Administrator Edit Transit

Notes:

1. Initially *Connected Sites* displays the current connected sites

(24) Administrator Create Transit



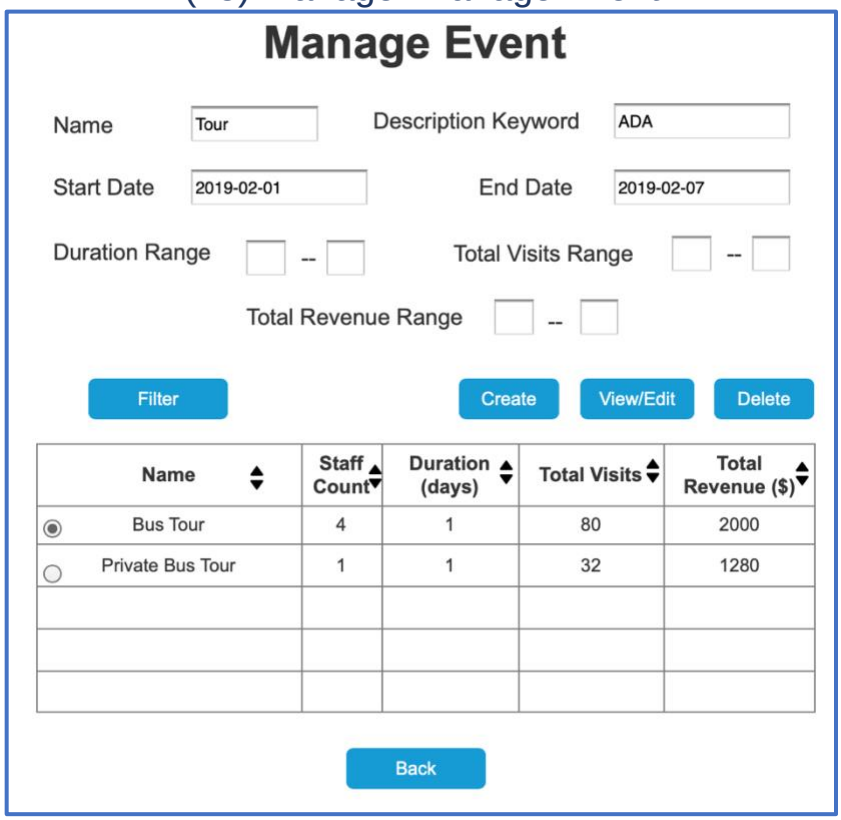
Screen 24 Administrator Create Transit

Notes:

2. All fields are required
3. The combination of *Transport Type* and *Route* is unique for each transit
4. *Transport Type* see Screen 15 User Take Transit Note 2
5. *Route* can be either a number or an alphanumeric string
6. Must select at least 2 *Connected Sites*
7. *Price* is nonnegative

Manager

(25) Manager Manage Event

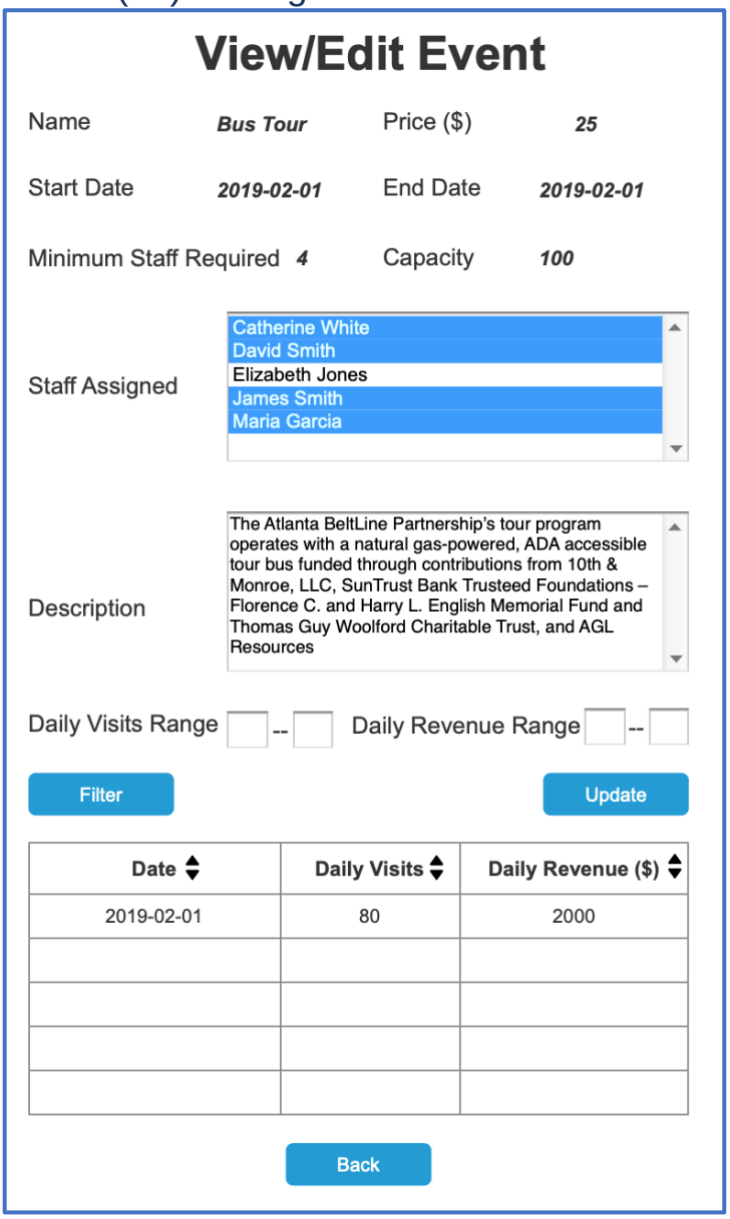


Screen 25 Manager Manage Event

Notes:

1. *Duration* is the number of days the event last (start and end on the same day means the event lasts for 1 day)
2. *Total Visits* is the number of visits logged to that event in this site
3. *Total Revenue* is calculated by multiplying number of visits and the event price
4. *Staff Count* is the number of distinct staffs assigned to that event in this site
5. Manager must select one existing event before deleting or editing it
6. Deleting an event will delete all the corresponding staff assignments and the logged event visits
7. Editing an event will change the event information for all the corresponding visits

(26) Manager View/Edit Event

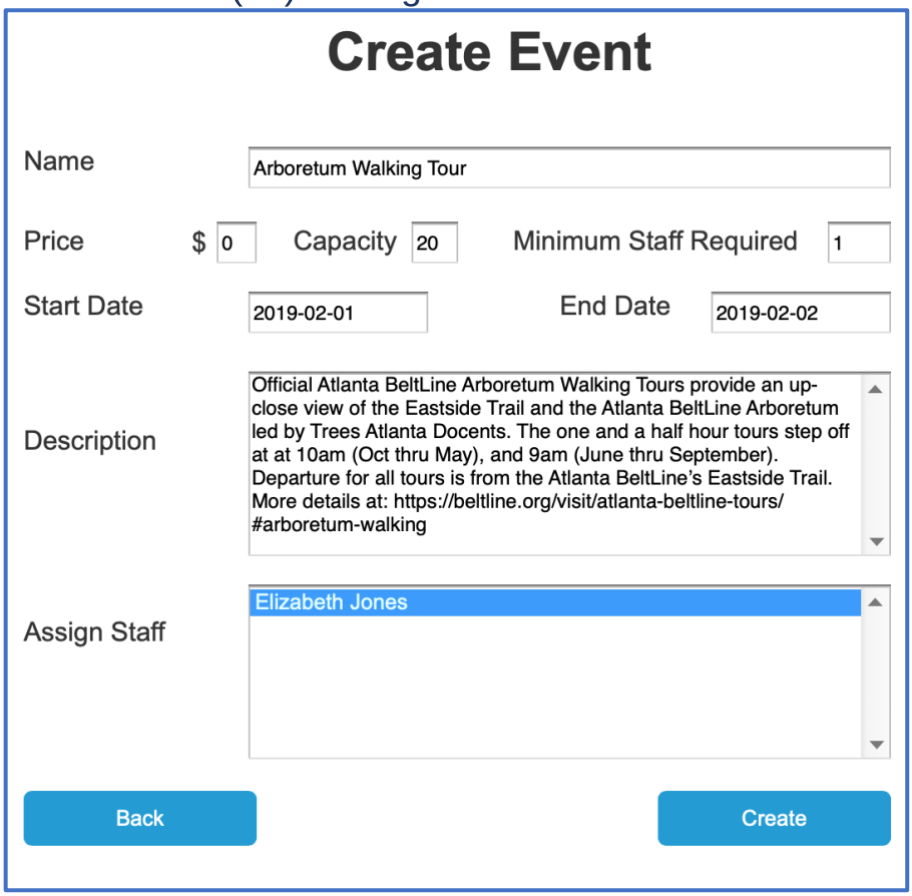


Screen 26 Manager View/Edit Event

Notes:

1. *Staff Assigned* and *Description* are editable
2. The number of *Staff Assigned* must not be fewer than the *Minimum Staff Required*
3. *Staff Assigned* displays the first name and last name of available staffs (staffs who are not assigned to other events during this event) as well as the staffs originally assigned to this event
4. The table displays daily results for the event from its start date to its end date
5. *Daily Visits* and *Daily Revenue* see Screen 25 Manager Manage Event Notes 2-3

(27) Manager Create Event

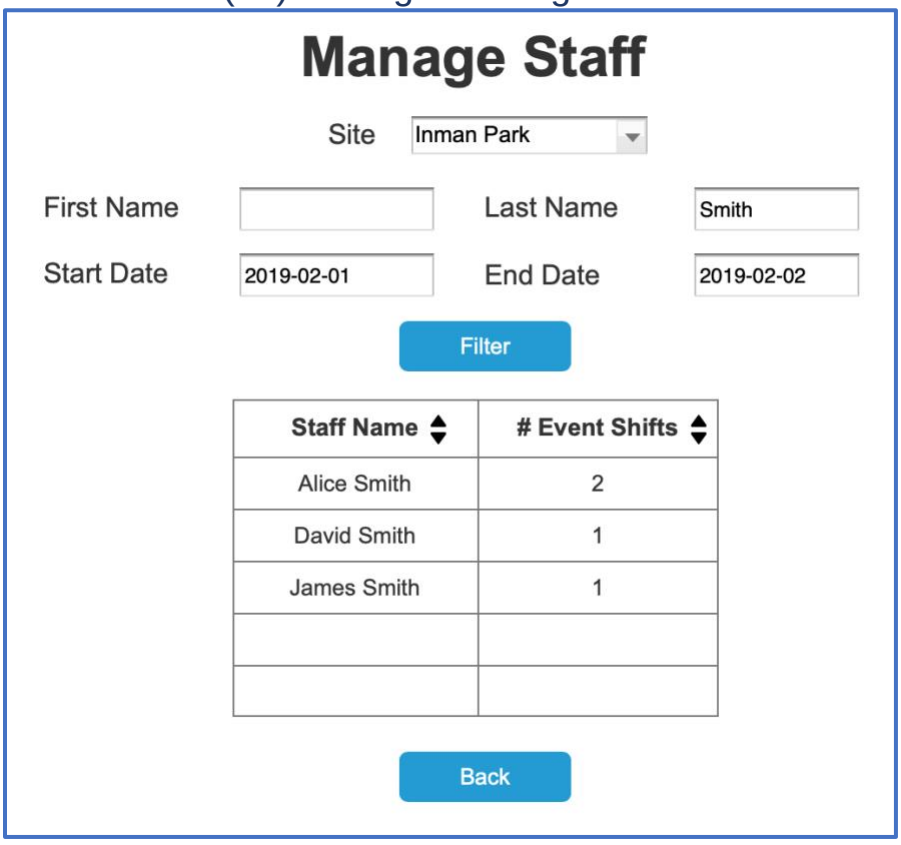


Screen 27 Manager Create Event

Notes:

1. All fields are required
2. Each event must be hosted at one specific site
3. The combination of *Event Name* and *Start Date* is unique for each site
4. Two events with the same *Name* in the same *Site* must not overlap (e.g. cannot have "Event One" in "Site One" from "2019-01-01" to "2019-01-10" and "Event Two" in "Site One" from "2019-01-10" to "2019-01-20" at the same time since they overlap—"2019-01-10").
5. *Price* is a nonnegative number
6. *Minimum Staff Required* is a positive integer
7. *Capacity* is a positive number
8. *Start Date* must come before *End Date*
9. *Assign Staff* displays the first name and last name of available staffs (staffs who are not assigned to any other events during this event's time period)
10. See Screen 26 Manager View/Edit Event Note 2
11. *Description* can be a long paragraph

(28) Manager Manage Staff

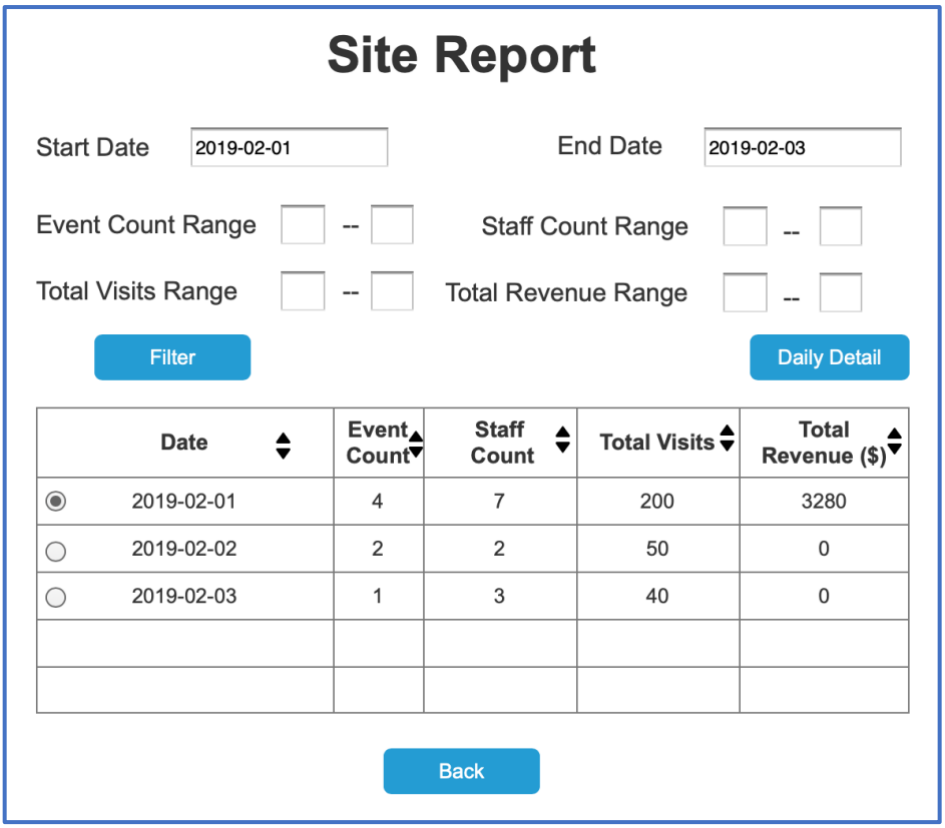


Screen 28 Manager Manage Staff

Notes:

1. *Staff Name* is the first and last name of the staff
2. *# Event Shifts* is the number of assigned events to that staff during the filtered date range

(29) Manager Site Report

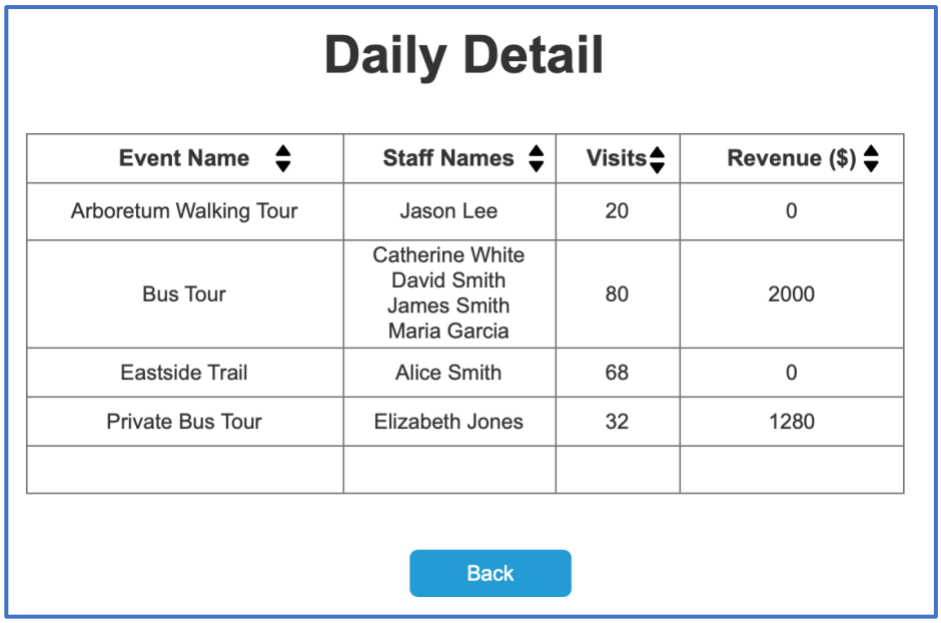


Screen 29 Manager Site Report

Notes:

1. Manager must fill in *Start Date* and *End Date*
2. Manager must select a *Date* before viewing the *Daily Detail*
3. *Total Visits* is the sum of visits to the site as well as the visits to all its events between *Start Date* and *End Date* entered
4. *Total Revenue* is calculated similar to Screen 25 Manager Manage Event

(30) Manager Daily Detail



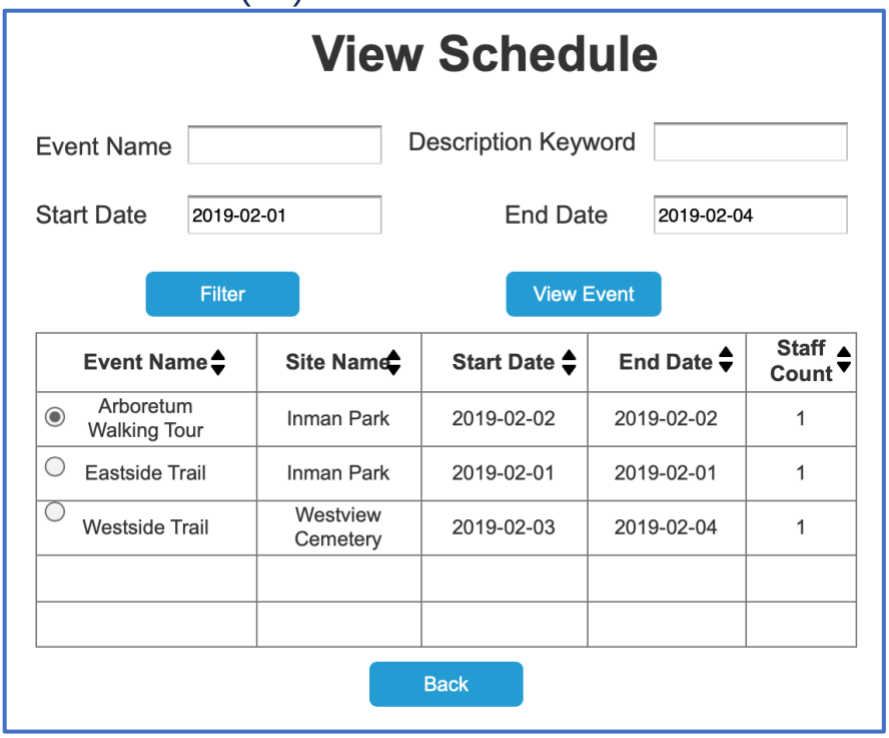
Screen 30 Manager Daily Detail

Notes:

1. *Staff Names* include the first and last names of all staffs who are assigned to the event during the *Start Date* and *End Date* (Screen 29 Manager Site Report) in alphabetical order of their first name
2. *Visits* and *Revenue* see Screen 25 Manager Manage Event Notes 2-3

Staff

(31) Staff View Schedule

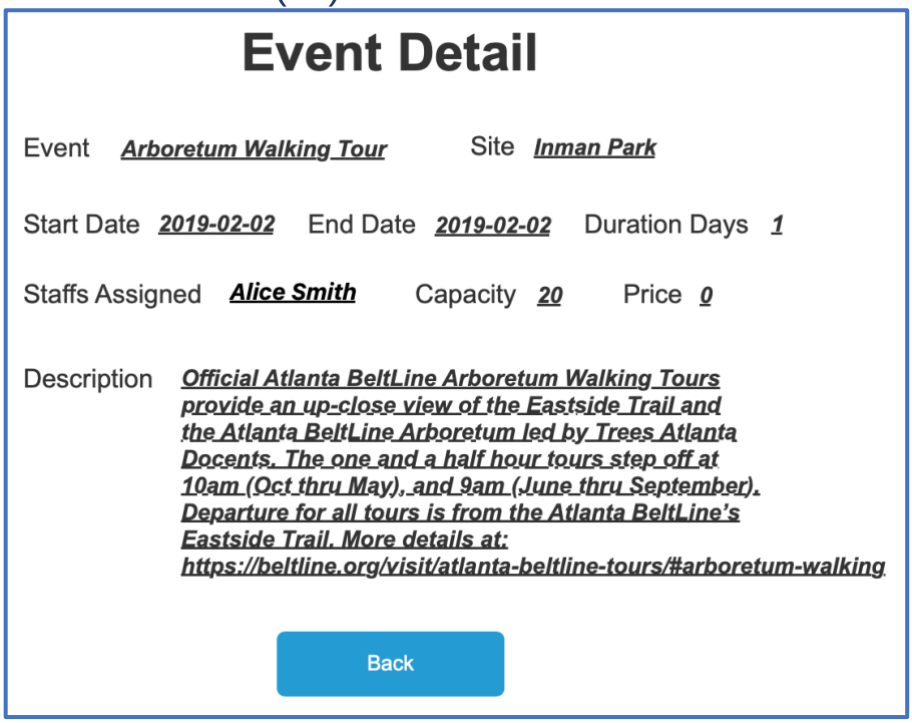


Screen 31 Staff View Schedule

Notes:

1. Staff must select an event before viewing its detail
2. If *Start Date or End Date* is filled, the result should contain events that start or end within the period of *Start Date* and *End Date* entered

(32) Staff Event Detail



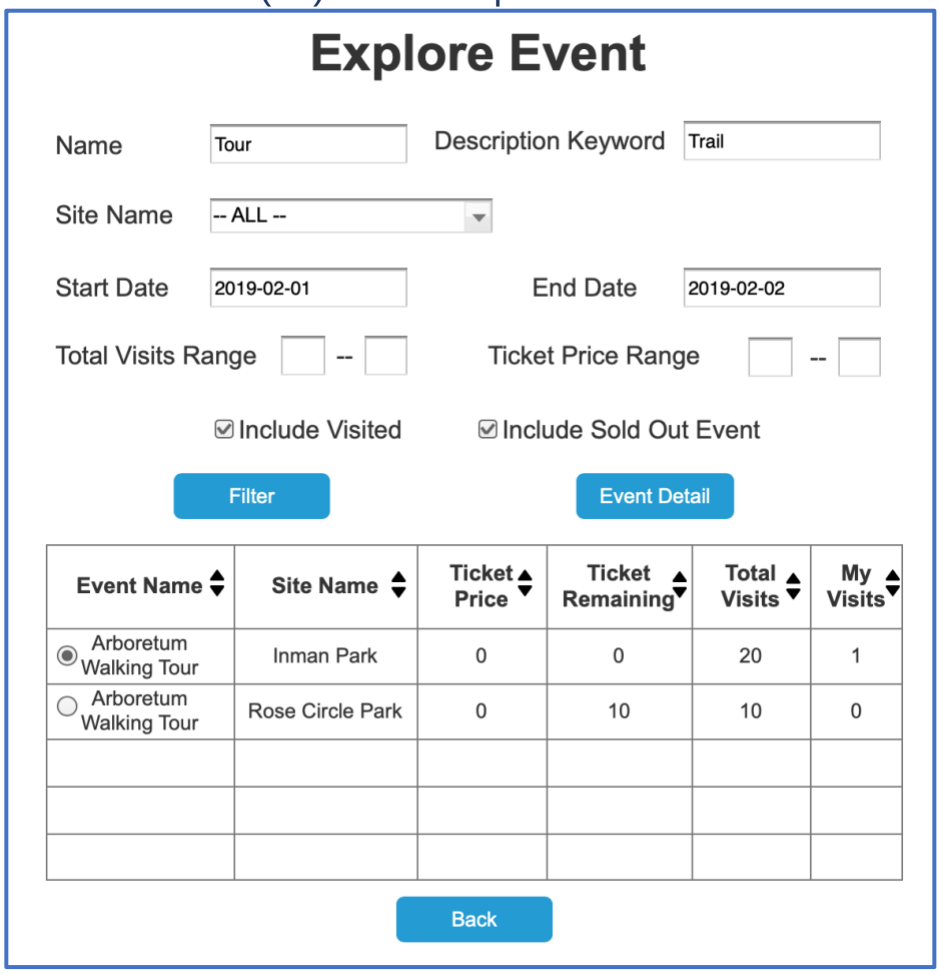
Screen 32 Staff Event Detail

Notes:

1. *Staffs Assigned* displays all the first and last names of staffs assigned to this event in alphabetical order

Visitor

(33) Visitor Explore Event

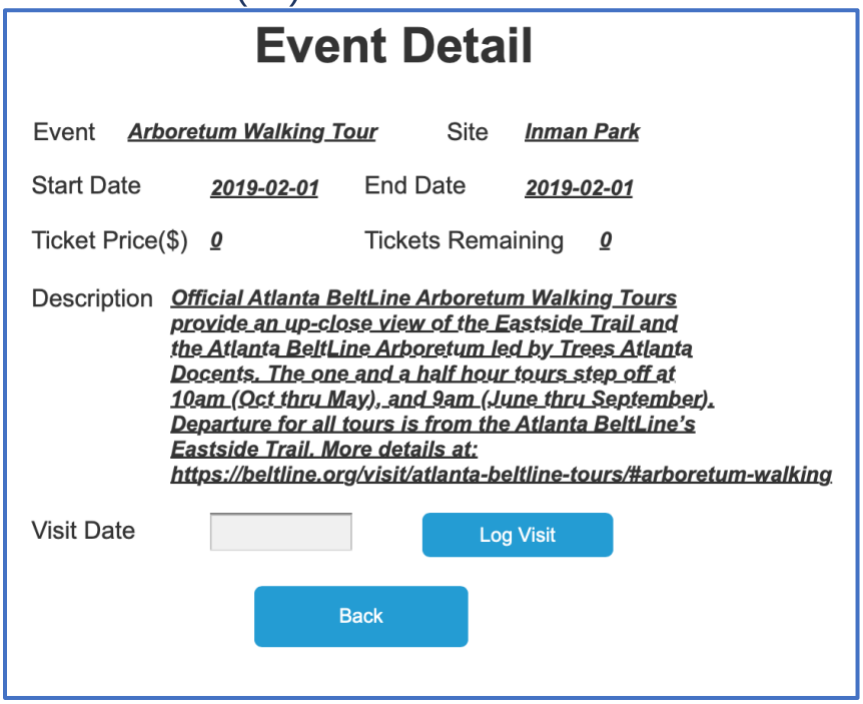


Screen 33 Visitor Explore Event

Notes:

1. *Site Name* is a dropdown list containing all the existing sites
2. *Ticket Remaining* is calculated from the difference between the *Event Capacity* and *Total Visits*
3. *My Visits* is the number of visits logged by the current visitor
4. Visitor must select an event before viewing its detail

(34) Visitor Event Detail



Screen 34 Visitor Event Detail

Notes:

1. Visitor can *Log Visit* to the selected event if there are still tickets remaining
2. *Visit Date* must be filled and must be within the *Start Date* and *End Date* before logging the visit
3. Visitor cannot log visit to the same event on the same date

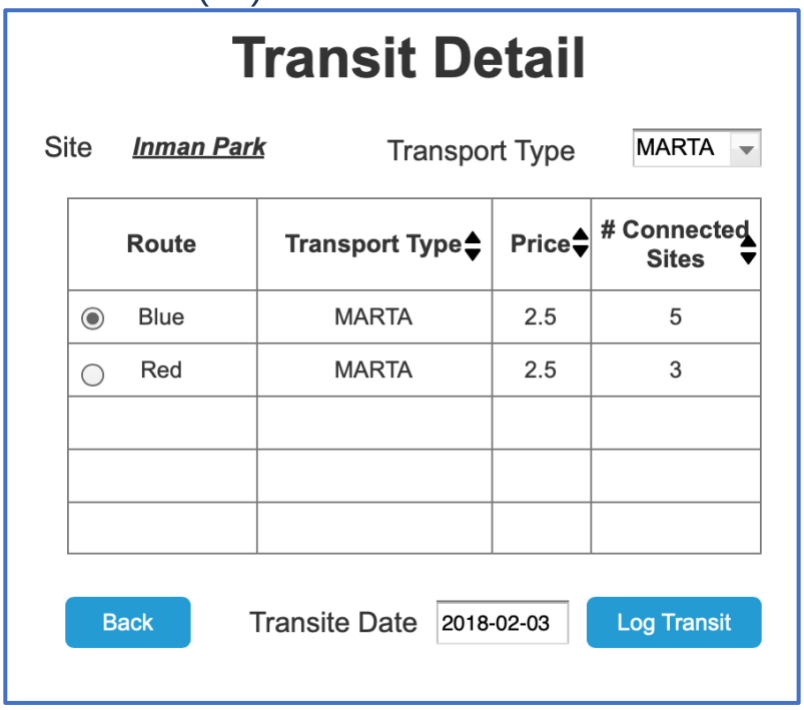
(35) Visitor Explore Site

Screen 35 Visitor Explore Site

Notes:

1. *Open Everyday* is a dropdown list containing “Yes,” and “No;” you can hardcode this list
2. Visitor must select a site before viewing its detail and its *transit* detail
3. *My Visits* is the sum of visits to the site as well as all of its events between the filled *Start Date* and *End Date*

(36) Visitor Transit Detail

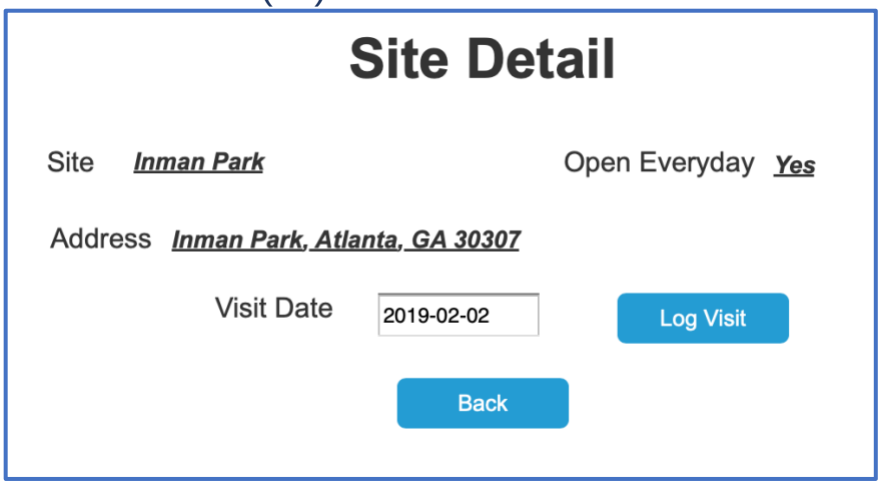


Screen 36 Visitor Transit Detail

Notes:

1. *Transport Type* see Screen 15 User Take Transit Note 2
2. Visitor must select a *Route* and fill in the *Transit Date* before taking the transit
3. *Transit Date* see Screen 15 User Take Transit Note 3

(38) Visitor Site Detail

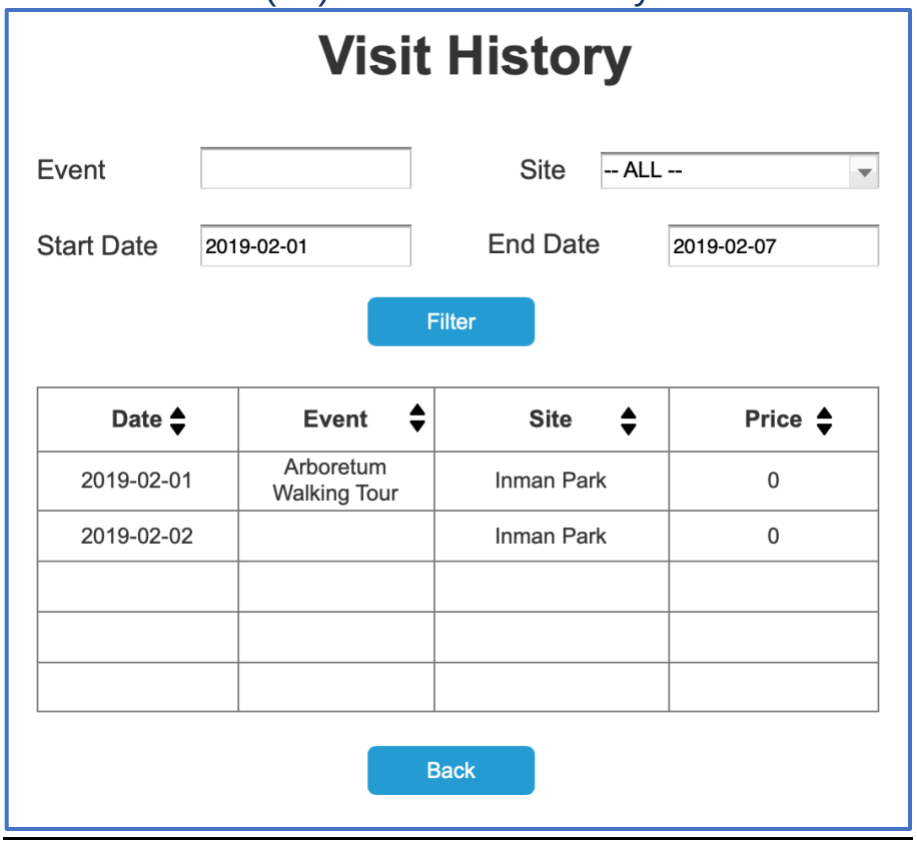


Screen 37 Visitor Site Detail

Notes:

1. Visitor cannot log visit to the same site on the same date (visiting an event in the site is different from visiting the site)

(37) Visitor Visit History



Screen 38 Visitor Visit History

Notes:

1. *Start Date* and *End Date* can be optional
2. *Price* is 0 if the event is free or if it is a site visit
3. *Event* column is empty if it is a site visit