Team Big Data

U-tification

Business Requirement Document

Date: 10/5/2022

Team Leader: Joseph Armas
Team Members: Joshua Gherman
Rhoy Oviedo
Frank Curry
Ghabrille Ampo
David DeGirolamo

Git Repository: https://github.com/JosephArmas/cecs-491A-Team-Big-Data

Version History

Current Version: Version 4

- Requirements
    - Added Events (Admin)
    - General Requirements
        - Adjusted Functional Requirements
    - Changed Formatting of Success Criteria For:
        - Litter Map
        - Pinning
        - Pinning (Admin)
        - Alerts
        - Alerts (Admin)
        - Reputation
        - Reputation (Admin)
        - Event Creation
        - Event Joining
        - Service Creation
        - Service Requests
        - Services (Admin)
        - Picture Upload
        - Picture Upload (Admin)
- Added new Reference

Previous Versions: V3, V2, V1

Table of Contents

Summary

U-tification aims to provide a litter map of litter, abandoned items, and vandalism that will have a management system in order to map places of high amounts of rubbish, track valid users, and increase awareness between places of garbage and local communities. In order to achieve our goal of beautification across California, we need to have a litter map that tells a cohesive story about local cleaning events and areas that are in need of cleaning services. Our goal is to have this website launched by May 10, 2023.

Requirements

Requirement
General Requirements
Description
Functional and Non-Functional Requirements that apply to the entire application.
Functional Requirements
1. Users exclusively have write access to their own profile or post. 2. Admins are able to access User data. 3. Admins are able to access all logged data. 4. Users are able to view all pin types. 5. The system must keep the user location to the respective user.
Non-Functional Requirements
1. All Features will be available using a web browser: Chrome 104.x (64 bit). 2. All Features are expected to have a *MTTRS* of 20 minutes following a system failure. 3. All Features should be available to 95% of California's residents every month during any time of day. 4. The system shall support from 1 to 15,000 concurrent users. 5. The system shall make clickable features clearly interactable. (mouse changes shape, buttons glow etc). 6. The system shall not crash due to Error Handling.
General Success Criteria
1. All text uses a standardized font. 2. All text will use *en-us.* 3. All text will be *legible.* 4. Users do not directly communicate with *Backend*. 5. Data is both accurately displayed on the Client side as well as Server side.

General Failure Criteria
Text flagged as invalid: Not standardized font, Not in en-us, Illegible. ● Error Handling: 1. No error message is displayed. 2. The website will reload itself. Data is not accurately recorded on Client and Server end. ● Error Handling: 1. Display message will be displayed: “Server Error. Reloading” 2. The website will reload itself. Data on both sides will reset as if the event did not occur. User communicates directly with the Backend. ● Error Handling: 1. Connection between User and Backend is severed immediately 2. Backend is reset to state prior to User interaction.

Feature
User Types
Description
Ensures the user's account has the authorization to do the action they want. Each user has their own individual account and is able to make any changes with proper authorization. Users are able to login to their account with their proper credentials and exit with a logout button.
Admin
1. Highest level user. 2. Able to perform actions that other users cannot. 3. Not bound by reputation. 4. Have read access to all users' information and interactions. 5. Can delete user's profile information from Server side. 6. Not able to create pins, create services, request services, or upload pictures. 7. Able to ban and unban user accounts. 8. Able to recover user accounts.
Reputable
1. Middle level user is able to perform all actions the lowest level can. 2. Authority to create events and request services. 3. Reputation must be greater than or equal to 4.2 rating.
Service
1. Low level user. Able to perform all actions the lowest level can. 2. Able to add business to list of businesses. 3. Able to receive notifications from user requests.
Regular
1. Lowest level user. Reputation is lower than 4.2 rating.

Feature
Litter Map
Description
Displays a map of hot spots for litter, abandoned items, and vandalism. This feature adds to the value of the project by being the source of any user's interaction and data logging.
Precondition
1. User must be Logged In. 2. User type is either: Admin, Reputable, Service, or Regular. 3. User has their Location turned on, so that the map may use it.
Functional Requirements
1. The system must allow a map of California to be displayed in order for the user to use it. 2. The system must allow the user the ability to move the map along the X-Axis and Y-Axis. 3. The system must allow the user to pan across the map. 4. The system must allow the user to use touch controls to pan across the map. 5. The system must allow the user to zoom in as close as 1 mile on the map. As in the zoom can get as close as rendering one square mile of the map. 6. The system must allow the user to zoom out as far as 100 miles on the map. As in the zoom can get as far out as rendering 100 square miles of the map. 7. The system must allow the user to view all of the currently active pins. 8. The system must allow the user to filter the pins based on *zip code* as well as distance from the user's current position. 9. The system must have a map with streets and parks with names so that users can identify locations. 10. The system must accept valid zip codes only from California.
Non-Functional Requirements
1. The system shall have the map load no longer than 10 seconds under any circumstances.

2. The system shall maintain at most 0.5 seconds of input lag with 10,000 visitors utilizing the feature.
3. The system shall have coordinates formatted as follows: 0.000000, 0.000000
4. The system shall have the ability to add additional areas to the map. Additional states could be added in the future.
5. The system shall have the ability to modify or remove areas of the map. In the case that they are no longer supported.
6. The system shall allow for the zoom out distance to be changed for a later date.
7. The system shall have the ability to add additional zip codes for future location additions i.e. zip codes for other U.S. States.
8. The system shall immediately display the last known zip location, after the user logged in.
9. The system shall have the litter map display take up half the size of the webpage.
10. The system shall allow the map to be full screened.

Successful Operation

1. The user is able to:
    a. Load the map.
    b. Move along the X-Axis and Y-Axis.
    c. Adjust the zoom.
    d. Pan across the map.
    e. Filter pins.
    f. View all pins.
2. The Map maintains at most 0.5 seconds of input lag when 10,000 visitors are utilizing the feature.
3. The system is able to:
    - Support 15,000 visitors.
    - Support new areas on the map.
    - Display User location upon Log In.
    - Have a sizable map.

Failed Operation

- The map takes longer than 10 seconds to load.
    - Error Handling:

1. The system should then inform the user that the map has timed out “Server Request Timeout. The Map Failed to Load”.
2. Then the system should Allow the map to be reloaded.

- The map has slow response time i.e. user moves map the map moves more than 0.5 seconds later.
    - Error Handling:

1. The system must add additional scale when overpopulated.

- The map does not display street names or parks.

1. The User is not notified.
2. The system shall allow the map to automatically reload.

- The map doesn’t zoom.

1. The User is notified and displays “Zoom Feature is Unresponsive”
2. The system shall allow the map to automatically reload.

- The map doesn’t pan.

1. The User is notified and displays “Pan Feature is Unresponsive”
2. The system shall allow the map to automatically reload.

Feature
Pinning
Description
An extension of the map feature where users are able to create pins on the map.
Precondition
1. A User must be Logged In. 2. A User is of type Regular, Service, or Reputable. 3. A User must be using the Map.
Functional Requirements
1. The system must allow users to place a pin anywhere on the Litter Map. 2. The system must allow users the ability to remove any pin that their respective account has placed. 3. The system must allow users the ability to modify the details of any pin that their respective account has placed. This includes title, description, pin type, and any attached images. 4. The system must allow the pin creator to add an image to the pin. 5. The system must allow the user to filter the pins by type: Litter pins, Group Event pins, Junk Item pins, Abandoned Item pins, and Vandalism pins. 6. The system must display the title and description of the event when the pin is accessed. 7. The system must have a way to let users exit the details of a pin. 8. The system must allow users to view the date, time, and location of created pins. Location details include zip code, global coordinates, and city. 9. The system must allow users to mark a pin as completed, removing it from the map. 10. The system must allow the user to view their cleaned or deleted pins in their User Profile. 11. The system must have the user confirm that they want to create a pin. 12. The system must log all pins created and deleted. The details included, the pin type, title, description, any images attached. 13. The system must accept valid zip codes only from California.

Non-Functional Requirements

1. The system shall allow a user to create a pin once every 3 minutes. Users are unrestricted by time of day.
2. The system shall handle 15,000 users posting pins with 3 seconds of latency.
3. The system shall be scalable to support up to 50,000 pins without the need of additional servers.
1. The system shall load the pins in no more than 15 seconds.
2. The system shall register the location of the pin and add it to the list of pins on the data store within 7 seconds.
4. The system shall have the ability to adjust the zip codes for future location additions.
5. The system shall display distance from the current location when the pin is clicked. Distance shown in miles.
6. The system must allow for additional pin types to be added in the future.
7. The system shall have the size of the pin be 40x40 pixels.
8. The system shall allow the title of pins a character limit of 30.
9. The system shall allow the description of pins a word limit of 150.
10. The system shall have pins of different types be shown in different colors.
    - Litter pins shall have a blue color: #1369f2.
    - Group Event pins shall have a yellow color: #f3f70f.
    - Junk Item pins shall have a green color: #07b30d.
    - Abandoned Item pins shall have a red color: #d11d17.
    - Vandalism pins shall have a purple color: #930ff2.

Successful Operation

The Pinning Feature is considered successful if:

1. The User is able to:
    a. View a Litter Map
    b. View pins displayed on the map.
    c. View pins shown on the map exactly where the pin was dropped.
    d. Have a confirmation notification when creating a pin.
    e. View that pins contain a title and description

2. The System is able to:
    - Show pins with an appropriate color
    - Show distance between the pin and the user.
    - Display pins 40x40 pixels.
    - Support 50,000 pins without additional servers.

Failed Operation

The Pinning Feature will fail when encountering:

- Pins are displayed inappropriately.
    - Error Handling:
    1. The User is notified of the error and displays “Error Displaying Pin”.
    2. The User is allowed to retry their pin request.
- A map is not viewable by a User.
    - Error Handling:
    1. The User is not notified.
    2. The map is automatically reloaded.
- Pins are shown with an invalid color.
    - Error Handling:
    1. The User is not notified.
    2. Invalid pins are reloaded to match data on the Server side.
- Users are unable to post pins.
    - Error Handling:
    1. The User is notified of the error and displays “Error! Unable to Post Pin. Please Try Again”.
    2. The User is able to retry posting the pin.
- Users are able to post pins more than once every 3 minutes.
    - Error Handling:
    1. The User is notified of an error and displays “Unable to Post. Users are Restricted to One Post Every Three Minutes.”.
    2. The most recent post by the User is removed.

Feature
Pinning (Admin)
Description
Part of the pinning feature that describes how Admins interact with the pins.
Precondition
1. An Admin must be Logged In. 2. A User is of type: Admin. 3. An Admin must be using a map or list of pins.
Functional Requirements
1. The system must allow Admin read access to all User's pin data. 2. The system must allow Admin the ability to modify or delete User's pin data in the case that a user's pin is inappropriate. In that it contains profanity, slurs, or crude images.
Non-Functional Requirements
1. The system shall take no longer than 7 seconds to delete the User's pin data. 2. The system shall ask for confirmation when modifying or deleting event data by stating "You are about to modify/delete a pin(s)." 3. The system shall always allow Admins to modify or delete pins 99 percent of the time every week.
Successful Operation
The Pinning Feature is considered successful if: 1. The Admin is able to: a. Remove User's data from the Server side. b. View a User's pin data
Failed Operation
● User's profile does not reflect Server side upon removal.

- Error Handling

1. Server side takes priority and will have its data reflected on the User Profile.
2. Admin is notified if their delete process went through and displays “Deletion was Successful”.

Feature
Alerts
Description
Notifications that inform users of updates to pins.
Precondition
1. User must be Logged In. 2. A User is of type: Regular, Service, or Reputable.
Functional Requirements
1. The system allows users to accept notifications from entered events. 2. The system allows users to accept notifications from pin drops by zip code. 3. The system allows users to accept notifications from pin drops by type: Litter pins, Group Event pins, Junk Item pins, Abandoned Item pins, and Vandalism pins. 4. The system allows users to ping other users when creating a pin, the ping only goes to those accepting notifications. 5. The system must have a method for the user where they can view a list of notifications. 6. The system must have Service Users receive notifications from User requests. 7. The system must allow the user to mark a notification as read. 8. The system shall allow each user to hold up to 100 notifications before they are automatically marked as read by the system. Only 1 notification will be marked as read automatically when the notification list is full, the oldest notification. 9. The system must accept valid zip codes only from California.
Non-Functional Requirements
1. All alert functions, with the exception of Admin alerts, are off by default. 2. The list of notifications marked as read will automatically be removed from the list 7 days after being marked.

3. The list of notifications must display all the notifications in no greater than 2 seconds.
4. The system shall allow the maximum number of notifications to be modifiable for future uses.
5. The notification list must be available to the users 95 percent of the time during each week PST.
6. The system shall have the ability to adjust the zip codes for future location additions.
7. The system shall display Notifications in Bold font.
8. The system shall display a list of notifications in chronological order by default.
9. The system shall inform users that their notifications list is full by stating “Notification inbox is full”.

Successful Operation

The Alert Feature is considered successful if:

1. The User is able to:
    a. Receive a notification about a pin they accepted notifications for and is able to view it.
    b. Not receive a notification for a pin they are not accepting notifications from.
    c. Ping other users when creating a pin.
    d. Have a notification inbox.
2. The system is able to:
    - Display a list of notifications within 2 seconds.
    - Displays notifications in Bold.
    - Notifies users of a full notification inbox.
    - Display a list of notifications in chronological order by default

Failed Operation

The Alert Feature will fail when encountering:

- The user never receives the notification that they were meant to receive.
    - Error Handling:
    1. The User is not notified of an error.
    2. The system shall automatically try to resend the notification.
- The user receives a notification for a pin they did not accept notifications for.
    - Error Handling:
    1. The User is notified of an error and displays “Previous notification is Invalid. Apologies for the False Flag”.

2. The notification is deleted from the User's profile.

- The pin creator's notification never actually deploys.
    - Error Handling:
    1. The User is notified of an error and displays "Notification Error. Pin Notification Not Deployed".
    2. The system shall automatically redeploy the notification.
- The list of notifications does not show the correct notifications.
    - Error Handling:
    1. The User is notified of an error "Error. Displayed List is Incorrect. Reloading".
    2. The list of notifications automatically is updated with information from Server side.
- The list of notifications does not load
    - Error Handling:
    1. The User is notified by stating "Unable to Load Notifications".
    2. The User is allowed to reload the notification list.
- The user receives a notification for the wrong pin type.
    - Error Handling:
    1. The User is notified and displays "Previous notification is Invalid. Apologies for the False Flag".
    2. The system shall update with User's preference.

Feature
Alerts (Admin)
Description
Part of the Alerts feature that defines how an Admin is able to control alerts.
Precondition
1. Admin must be Logged In. 2. User type is Admin.
Functional Requirements
1. The system allows Admins to send notifications to Users.
Non-Functional Requirements
1. The notifications are the same as the notifications of a pin, but distributed to all applicable users. 2. The system shall be scalable enough to send a notification to at least 20,000 users at the same time and have users receive it within 20 seconds of sending the notification.
Successful Operation
The Alert Feature is considered successful if: 1. The Admin is able to: a. Send a notification to all applicable Users and they received it. 2. The system is able to: ● Send a notification to 20,000 users at once with 20 seconds of latency.
Failed Operation
The Alert Feature will fail when encountering: ● The Admin notification is not sent to the correct users, or never sent at all. ○ Error Handling:

1. The Admin is notified and displays “Notification Error. Please Retry Alert”
2. The Admin may retry sending the Alert.

Feature

Reputation

Description

A feature of the system that gives all users a rating that is adjusted by interacting with pins and other users.

Precondition

1. User must be Logged In.
2. A User is of type: Regular, Service, or Reputable.

Functional Requirements

1. The system must allow users to have a reputation rating from zero to five as a decimal value. Example: 3.0, 1.3, 4.8.
2. The system must allow all users to view User reputation.
3. The system must allow for reports to impact their reputation, either negatively or positively.
4. The system must have a way to report other users, not the user's own profile. The user reports rate the user from zero to five as a decimal value.
5. The system must allow for the user to view reports submitted by the respective user.
6. The system must keep the reputation from being altered on the browser. The user reputation is kept on the servers to prevent any kind of user tampering.
7. The system must allow users to gain reputation by creating, the reputation gained is 0.1 per pin only up to a threshold of 3.5.
8. The system must allow users to gain reputation by marking pins as completed, the reputation gained is 0.1 per pin completed only up to a threshold of 3.5.
9. The system must log all changes to the reputation.
10. The system shall recognize a reputable user as a user with a 4.2 or greater rating.
11. The system shall start every user with a rating of 2.0.
12. The system shall adjust the users rating by sum score divided by the number of reviews. (sum score / number of reviews)

Non-Functional Requirements

1. The system shall display the rating as a ratio out of five “stars” along with the decimal rating.
2. The system shall retrieve other user reputation no later than 0.5 seconds after requested.
3. The system shall retrieve a user's own reputation no later than 0.3 seconds after requested.
4. The system shall notify a User when they change from Regular user to a Reputable user.
5. The system shall support up to 5,000 reports on a user’s reputation to calculate rating.
6. The system shall support the reputation for all users with the exception of Admin users.
7. The system shall load the list of reports sent by the user no later than 0.5 seconds after being requested.
8. The system shall allow the value reputation gain from creating or marking pins as completed to be modifiable so that it may be adjusted in future updates. The threshold shall also be modifiable so it may be adjusted in future updates.
9. The system shall allow the reputable user minimum rating to be modifiable so that it may be adjusted in future updates.
10. The system shall allow the user rating formula to be modifiable.
11. The system shall continue to retrieve user reputation information optimally even with 100 users requesting the information.
12. The system shall update user reputation after gaining or losing reputation no later than 5 seconds after it was sent to the servers.

Successful Operation

The Reputation Feature is considered successful if:

1. A User is able to:
    a. Successfully report another User positively, the other User’s reputation increases.
    b. Successfully report another User Negatively, the other User’s reputation decreases.
    c. View the reputation of other Users.
    d. Gain reputation from creating and marking pins as completed.
2. The system is able to:
    - Display a User’s rating as a ratio of 5 “stars”.
    - Support 5,000 ratings on a User.
    - Retrieve User’s information within 0.5 seconds of a request.

Failed Operation
The Reputation Feature will fail when encountering: ● The user is unable to view the reputation of other users. ○ Error Handling: 1. The user is notified of an error displaying “Reputation is Unavailable Right Now. Please Try Again.” 2. The user is able to attempt viewing reputation again. ● The user’s report does not impact the other user’s reputation. ○ Error Handling: 1. The user is notified of an error displaying “Report Failed to Submit. Please Try Again.” 2. The user is able to attempt reporting the other user again. ● The user’s report is not received by the server. ○ Error Handling: 1. The user is not notified of an error 2. The server is automatically reloaded. ● The user’s “stars” do not match the rating. ○ Error Handling: 1. The user is not notified. 2. The UI automatically tries to reload. ● A Regular user is not changed into a Reputable user upon reaching 4.2 star rating. ○ Error Handling: 1. The User is not notified of the error. 2. The system will automatically update the User's type.

Feature
Reputation (Admin)
Description
An extension of the Reputation feature that defines how the Admin user type is able to control user reputation.
Precondition
1. Admin must be Logged In. 2. User type is Admin
Functional Requirements
1. The system must allow Admins to reset the rating of other users. 2. The system must allow Admins the ability to modify or delete user feedback.
Non-Functional Requirements
1. The system shall process modification requests within 7 seconds. 2. Users are notified about Admin modifications to their profile.
Successful Operation
The Reputation Feature is considered successful if: 1. The Admin is able to: a. Reset ratings of users. b. Modify or delete user feedback. 2. The system is able to: ● Notify users of Admin activity on their account. ● Have Admins modify the requirements for a User gaining reputation.

Failed Operation
The Reputation Feature will fail when encountering: ● User's ratings are not reset/ User's feedback is not modified or deleted. ○ Error Handling: 1. Admin is notified about the failed operation and displays "Modification to User Failed. Please Try Again". 2. Admin is able to retry their request. ● User does not receive a notification ○ Error Handling: 1. The User is not notified. 2. The system will automatically try to resend the original notification.

Feature

Event Creation

Description

A derivative of the events feature, it defines how reputable users are able to create events.

Precondition

1. Reputable user must be Logged In.
2. A User is of type: Reputable.
3. A User must be viewing the map so that the pin may be set.

Functional Requirements

1. The system must allow users to create events by placing a pin on the map.
2. The system must allow the pin to have a title and a description so users can understand the purpose of the event.
3. The system must allow the pin creator to edit the title and description in case any typos or details have changed.
4. The system must allow the pin creator to upload a photo along with the event pin.
5. The system must allow the user to cancel the creation of the event pin before posting the pin.
6. The system must allow the pin creator to toggle the display of attendance for an event.
7. The system must allow the event to be logged.

Non-Functional Requirements

1. The system shall allow the title of pins a character limit of 30.
2. The system shall allow the description on pins a character word limit of 150.
3. The system shall place the pin on the map as soon as the event is created.
4. The system shall allow a user to post an event once every 7 days.
5. The feature must create an event 100 percent of the time.
6. The availability of the feature must be available 100 percent of the time that the application is

functional.

7. The system shall be extensible to include new inputs for the event pin such as a banner.
8. The system shall verify that the user is of user type reputable.

Successful Operation

The Event Creation Feature is considered successful if:

1. A Reputable User is able to:
    a. Create an event pin.
    b. Add a title and description to an event pin.
    c. Modify a pin or cancel a pin request.
    d. Post a pin once every 7 days.
2. The system is able to:
    - Accurately display event information.
    - Have event pins be extensible for future additions.

Failed Operation

The Event Creation Feature will fail when encountering:

- The event is created but the pin does not appear on the map.
    - Error Handling:
    1. The User is not notified.
    2. The system will automatically update itself.
- The event is created but lacks a title and description.
    - Error Handling:
    1. The User is notified and displays “Trouble Displaying Descriptors Please Try Again”.
    2. The User is able to modify their pin and repost.
- The event is edited by another user that is not the event creator.
    - Error Handling:
    1. The User is notified and displays “An Error Occurred with Your Pin”.
    2. The system shall restore pin to what it was prior to other user intervention.
- The toggle for attendance does not update (turn on/ turn off).
    - Error Handling:

    1. The User is notified and displays “Error Updating Attendance Display Preference”
    2. The system shall automatically update the attendance preference.

- A Non-Reputable type user is able to create an event.
    - Error Handling:

1. The User is notified and displays “Event Creation is Inaccessible to Non-Reputable Users”.
2. The system must delete the created event.

Feature
Event Joining
Description
Defines how a user can join an event pin.
Precondition
1. User must be Logged In. 2. A User is of type: Regular, Service, or Reputable. 3. A User must be viewing a Group Event pin.
Functional Requirements
1. The system must allow users to join an event. 2. The system must allow users to leave an event. 3. The system must allow the user to view their joined events in their User profile. 4. The system must allow 1 to 100 users to join an event. 5. The system must log the users joining and leaving events.
Non-Functional Requirements
1. The system shall inform the user that they have joined the event, stating “You have joined the event”. 2. The system shall inform the user that they have left the event ,stating “You have left the event”. 3. The system shall take no longer than 7 seconds to process a user joining an event. 4. The system shall take no longer than 7 seconds to process a user leaving an event. 5. The system shall give confirmation of joining an event no later than 7 seconds after being requested. 6. The system shall give confirmation of leaving an event no later than 7 seconds after being requested. 7. The system shall allow the maximum number of users to be modifiable so that it can be adjusted in a later update. 8. The system shall allow users to join or leave an event until the event pin is deleted or marked as cleaned up.

Successful Operation
The Event Joining Feature is considered successful if: 1. A User is able to: a. Join an event. b. Leave an event. c. View joined events on their User profile. 2. The system is able to: ● Record a User as having joined the event within 7 seconds on the client and Server side. ● Record a User as having left the event with 7 seconds on the client and data store side. ● Have 100 Users join an event.
Failed Operation
The Event Joining Feature will fail when encountering: ● The system took longer than 7 seconds to process a user joining an event. ○ Error Handling: 1. Inform the user that the process has timed out “Process Timed Out. Please Try Again”. 2. Allow the user to attempt the join process again. ● The system does not log joining or leaving events. ○ Error Handling: 1. User is not informed of error. 2. The system shall automatically update itself. ● The system allows more than 100 users to join an event. ○ Error Handling: 1. Users that join after the initial 100 are notified of their removal from the event and displays “Unable to Join Event. Attendance Limit Has Been Met”. 2. The system removes users from being logged as participating in the event.

Feature
Events (Admin)
Description
Defines how the admin interacts with the events feature.
Precondition
1. Admin must be Logged In. 2. User type is Admin
Functional Requirements
1. The system must allow Admins to read user event data. 2. The system must allow Admin the ability to modify or delete User's event data in the case that a user's event is inappropriate. In that it contains profanity, slurs, or crude images.
Non-Functional Requirements
1. The system shall take no longer than 7 seconds to delete User's event data. 2. The system shall allow for multiple events to be deleted at once. 3. The system shall ask for confirmation when modifying or deleting event data by stating "You are about to modify/delete an event(s)." 4. The system shall always allow Admins to modify or delete events 99 percent of the time every week.
Successful Operation
The Event Feature is considered successful if: 1. The Admin is able to: a. Remove User's data from the Server side. b. View a User's event data.

Failed Operation
The Event Feature will fail when encountering: ● User's profile does not reflect Server side upon removal. ○ Error Handling 1. Server side takes priority and will have its data reflected on the User Profile. 2. Admin is notified if their delete process went through and displays "Deletion was Successful".

Feature
Service Creation
Description
Defines how a user is able to create a service to assist with clean-up
Precondition
1. User must be Logged In. 2. A User is of type: Regular or Reputable. 3. User must already own a service.
Functional Requirements
1. The system must allow users the ability to add their services to the application. 2. The system must allow the user to enter their company name, description, phone number, and website when adding their services. 3. The system must allow the user who created a service, the ability to remove the service. 4. The system must allow the Service user an option to limit services based upon distance from location or zip code. 5. The system must allow the user to select the types of pins they would provide services for. 6. The system must allow only 10,000 Service users. 7. The system must tie only one service to one Service user. 8. The system must accept valid zip codes only from California.
Non-Functional Requirements
1. The system shall accept a company name of 30 characters as a valid entry. 2. The system shall accept a description with a 200 word limit as a valid entry. 3. The system shall accept a phone number of 10 digits as a valid entry. ● All valid phone numbers will have a country code of +1. 4. The system shall accept any input as a valid link to a website.

- If the line is left empty, the system shall pass the website with “N/A”.

5. The system shall have only one service per User account.
6. The system shall have the ability to adjust the zip codes for future location additions.
7. The system shall take no longer than 5 seconds to process a service creation.

Successful Operation

The Service Creation Feature is considered successful if:

1. A User is able to:
    a. Successfully register their service.
    b. Remove their service.
    c. Choose what pins their service will provide for.
    d. Link one service to their account.
2. The system is able to:
    - Accept valid company names and descriptions
    - Accept valid Website links

Failed Operation

The Service Creation Feature will fail when encountering:

- The Service user removes their service from the system but it stays in the list of services.
    - Error Handling:
    1. The User is not notified of an error.
    2. The system shall automatically update.
- The User adds their service to the system but it does not appear in the list of services.
    - Error Handling:
    1. The User is not notified of an error.
    2. The system shall automatically update.
- The service is available outside the distance stated by the user.
    - Error Handling:
    1. The User is notified and displays “Display Error. Some Services are Outside your Limits.”
    2. The system shall automatically update.
- The service’s entered company name is longer than 30 characters.

- Error Handling:
  1. The User is notified and displays “Syntax Error. Title is over 30 Character Limit.”
  2. The User is allowed to reenter the title of the service.

- The service’s entered description is longer than 200 words.
  - Error Handling:
  1. The User is notified and displays “Syntax Error. Description is over 200 Word Limit.”
  2. The User is allowed to reenter the description for the service.

- The Service User’s account is not updated correctly, resulting in the User still being viewed as Regular by the system.
  - Error Handling:
  1. The User is not notified.
  2. The system shall automatically update.

Feature
Service Requests
Description
Defines how users are able to request a service for their pin.
Precondition
1. User must be Logged In. 2. A User is of type Reputable, Service, or Regular. 3. A User must have a created, active pin.
Functional Requirements
1. The system must allow users to view a list of cleaning services. 2. The system must allow users to request a service for their created pin. 3. The system must allow the user that requested a service to cancel said service. 4. The system must allow for the Service user to accept a request. 5. The system must allow for the Service user to deny a request. 6. The system must allow the Service user to view their requests, both accepted and pending, allowing the Service user to see the location of the pin. 7. The system must allow the service to be recommended or not recommended by the user who requested the service. 8. The system shall inform the Service user that request is pending. 9. The system must accept valid zip codes only from California.
Non-Functional Requirements
1. The system shall display a list of cleaning services within the limit of zip code or specified distance. ● Users may specify a distance up to 20 miles away to find cleaning services. 2. The system shall take no longer than 7 seconds to create a list of cleaning services. 3. The system shall inform the user that the request has been sent by stating “Request sent”.

4. The system should send notifications no later than 0.4 seconds after a request is received by the server.
5. The system shall allow a user to cancel a service up to 3 hours before scheduled cleaning.
6. The system shall allow Service users to accept multiple requests that have conflicting time with other requests.
7. The system shall allow Service users to have a limit of 25 accepted requests at one time. Completing a request or having a request be canceled removes it from the list of 25 accepted requests.

Successful Operation

The Service Request Feature is considered successful if:

1. A User is able to:
    a. View a list of cleaning services and order a cleaning service to their pin.
    b. Cancel a service up to 3 hours before the scheduled cleaning.
    c. Have a Service User informed of their pending request and have it be viewed.
    d. Have a Service User deny their request or accept.
    e. Recommend the Service
2. The system is able to:
    - Create a list of cleaning services within 7 seconds of the request.
    - Inform a User of a sent request.
    - Have service users accept up to 25 request at once.

Failed Operation

The Service Request Feature will fail when encountering:

- The service was unable to be processed.
    - Error Handling:
    1. Inform the user that the request was not processed “ Runtime Error. Service Process Failed to be Requested”.
    2. The system shall prompt the User to retry the request.
- The system takes longer than 7 seconds to create a list of services.
    - Error Handling:
    1. The User is notified and displays “Runtime Error. Unable to Display List. Reloading.”.

2. The system shall automatically reload list of services.

- The system displays services further than specified by the user or more than 20 miles away.
    - Error Handling:
    1. The User is notified and displays “Display Data is Incorrect. Reloading.”
    2. The system shall automatically update displayed services with correct information.

Feature
Services (Admin)
Description
Defines how Admins are able to interact with service features.
Precondition
1. Admin is Logged In. 2. User type is Admin.
Functional Requirements
1. The system allows Admins to see which service a user recommended. 2. The system allows Admins to see the number of recommended services by a user. 3. The system allows Admins to modify or delete a service in case of the service no longer being used, the service is using malpractice, or service details contain vulgar content.
Non-Functional Requirements
1. The system shall allow an Admin to sort services.
Successful Operation
The Services Feature is considered successful if: 1. An Admin is able to: a. Modify a service. b. Delete a service. c. View recommendation of a service. 2. The system is able to: ● Allow services to be sorted.

Failed Operation
The Services Feature will fail when encountering: ● Admin is unable to delete a service. ○ Error Handling: 1. The Admin is notified and displays “Unable to Delete Service. Try Again” 2. The system shall allow the Admin to attempt deleting the service again. ● Admin is unable to successfully modify a service. ○ Error Handling: 1. The Admin is notified and displays “Error. Modification was Unsuccessful. Please Try Again.” 2. The system shall allow the Admin to retry modifying the service. ● Admin is unable to see the recommendation of a service. ○ Error Handling: 1. The Admin is notified and displays “Display Error. Reloading” 2. The system shall automatically reload the service.

Feature

Picture Upload

Description

Picture Upload allows users to upload photos to their pins, as well as store the images on their account.

Precondition

1. User must be Logged In.
2. A User is of type: Regular, Service, or Reputable.
3. User is viewing their created pin.

Functional Requirements

1. The system must allow users to upload an image on a pin.
2. The system must allow users to look at all the pictures they have uploaded to the system.
3. The system must allow the user to control read and write access.
4. The system must allow the user only to upload to their respective account.
5. The system must allow a user to upload an image to their user profile as a profile image.

Non-Functional Requirements

1. The system shall consider only photos of type .Jpeg and .png as valid.
2. The system shall consider only photos that don't exceed 9 megabytes as valid.
3. The system shall display a pop-up notification displaying "Invalid Photo Type" when a photo with an invalid type is uploaded.
4. The system shall display a pop-up notification displaying "Photo size is too big" when a photo over 9 megabytes is uploaded.
5. The system shall display both invalid notifications when both criteria on photos have been flagged as invalid.
6. The system shall work optimally with more than 100 images.
7. The system shall be extensible to include valid image sizes of images from up-to-date phones.

Successful Operation

The Picture Upload Feature is considered successful if:

1. A User is able to:
    a. Upload a picture.
    b. View pictures that they have uploaded.
    c. Change profile image with an uploaded photo.
    d. Delete an image they posted.
    e. Have an image viewed by other users.
2. The system is able to:
    - Accept Photos of valid format (Size and Type).
    - Works optimally with more than 100 images.
    - Be extensible to include popular formatting and file sizes in the future.

Failed Operation

The Picture Upload Feature will fail when encountering:

- A User is able to upload an image to a pin that is not theirs.
    - Error Handling:
    1. The User is notified and displays “Posting Issue. Invalid Action Performed.”
    2. The system shall automatically reverse actions performed by User.
- A User uploads an image that is not in .jpeg or .png format.
    - Error Handling:
    1. The User is notified and displays “Invalid Format. Accepted Images are .JPEG or .PNG.”
    2. The system shall remove uploaded image and allow User to retry.
- A User uploads an image larger than 9 megabytes.
    - Error Handling:
    1. The User is notified and displays “File Size Too Big. Accepted Sizes are 9MB and Less”
    2. The system shall remove uploaded image and allow User to retry.
- A User is unable to access their images.
    - Error Handling:
    1. The User is notified and Displays “Error. Cannot Access Images. Please Try Again.”

2. The system shall automatically reload images.

Feature

Picture Upload (Admin)

Description

Defines how Admins are able to interact with user's uploaded images.

Precondition

1. Admin is Logged In.
2. User type is Admin.

Functional Requirements

1. The system must allow Admin to delete User's uploaded images in case of crude content.

Non-Functional Requirements

1. The system shall display an uploaded image to Admin after at most 7 seconds.
2. The system shall work optimally with more than 100 images.

Successful Operation

The Picture Upload Feature is considered successful if:

1. An Admin is able to:
    a. Delete a user's image.
    b. View 100 images within 7 seconds of loading.

Failed Operation

The Picture Upload Feature will fail when encountering:

- Admin is unable to delete images.
    - Error Handling:
    1. The Admin is notified and displays "Error. Unable to Delete. Please Try Again"
    2. The system shall allow the Admin to retry deleting the images.
- Image takes more than 7 seconds to load.

- Error Handling:
  1. The Admin is notified and displays “Runtime Error. Reloading.”
  2. The system shall automatically reload the image.

- System slows due to too many images.
  - Error Handling:
  1. The Admin is not notified
  2. The system shall automatically reload.

Glossary

Backend - Part of the computer system that deals with storing data. Synonym: Server Side

En-us - This annotation refers to the English language using United States colloquialism.

Legible - The text displayed can be easily read. Not jumbled

MTTRS - The mean time to restore the system.

Zip code - In this document zip code refers to valid zip codes in California. Valid zip codes for California range from 90001 to 96162 [1].

References

[1] "Zip-codes.com," *Listing of all Zip Codes in the state of California*. [Online]. Available: https://www.zip-codes.com/state/ca.asp. [Accessed: 29-Sep-2022].

[2] *A guide to functional requirements (with examples)*. [Online]. Available: https://www.nuclino.com/articles/functional-requirements. [Accessed: 29-Sep-2022].

[3] J. Weller, "Download free BRD templates," *Smartsheet*, 18-Nov-2021. [Online]. Available: https://www.smartsheet.com/content/business-requirement-document-templates. [Accessed: 01-Oct-2022].

[4] "Non-functional requirements: Examples, types, how to approach," *AltexSoft*, 12-Feb-2020. [Online]. Available: https://www.altexsoft.com/blog/non-functional-requirements/. [Accessed: 01-Oct-2022].

[5] *Creating custom map marker pins for Google Maps, expressmaps and Mapbox*. [Online]. Available:https://www.storelocatorwidgets.com/supportarticle?title=Creating+custom+map+marker+pins+for+Google+Maps%2C+ExpressMaps+and+Mapbox#. [Accessed: 04-Oct-2022].

[6] V. Vong, "CSULB/SAMPLE-REQUIREMENTS.PDF at master · V-vong3/CSULB - github." [Online]. Available: https://github.com/v-vong3/csulb/blob/master/cecs_491/docs/sample-requirements.pdf. [Accessed: 05-Oct-2022].