Circuit Circus Project: System Design

Team Members: Reed Blazek, Allison Gigax, Keenan Herbe, & Nick Reiner

- Project Description
    - E-waste poses a large growing issue. With the popularity of shorter lifespan, knock off electronics, more E-waste is being created than ever. In 2022, around 50 million tons of E-waste were reported, and the trend only points upwards. A substantial portion of these landfill-bound devices are only partially broken and have potential for partial or entire salvage. These same parts are sought after by hobbyists, DIYers, and other individuals but they are forced to buy them new - increasing cost and waste. Circuit Circus aims to reduce e-waste by making parts destined for landfills more accessible to buyers.

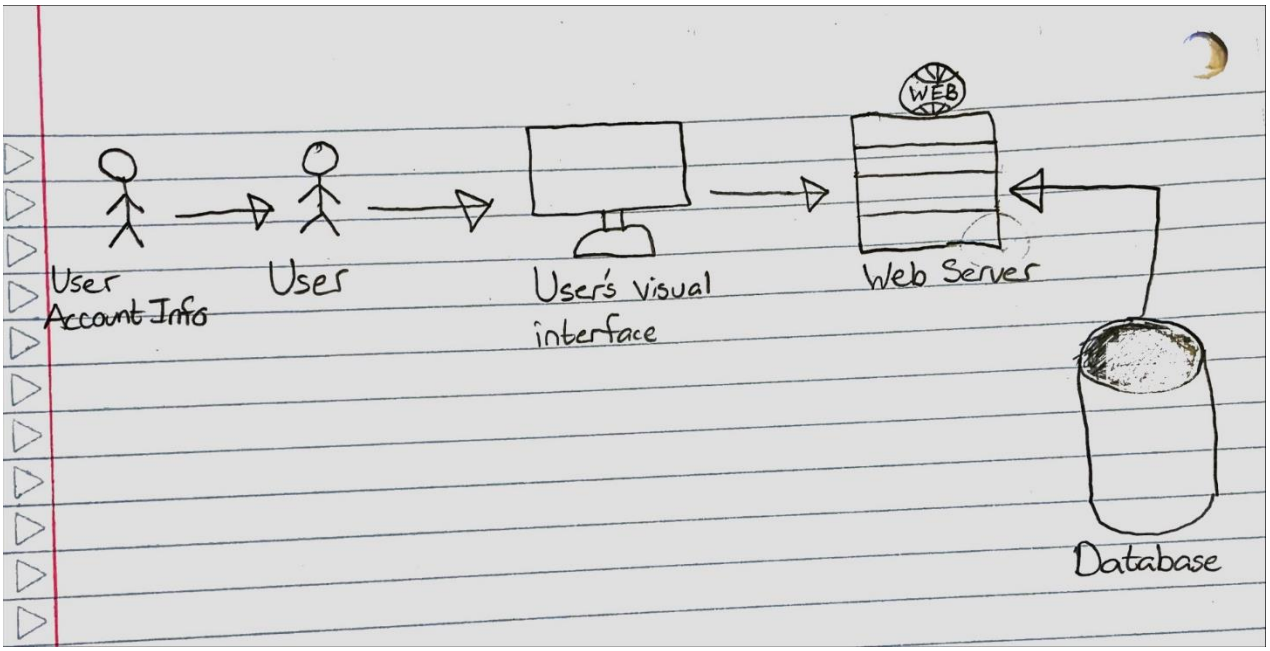
- Solution Description
    - Circuit Circus is a reselling platform with the objective of reusing and repurposing e-waste. The website enables users to engage in two primary activities: selling old or damaged electronic products/components and purchasing affordable electronics products/components for repair or repurposing purposes. To enhance the user experience, the platform offers filtering options and personalized search functionality for specific components, microelectronics, and materials. The search terms provided are a result of a combination of information provided by sellers and data sourced from OEMs (original equipment manufacturers). The website also allows users to manage their account details and interact with other users through the public forum and instant messaging.

- Development Guidelines
    - Prototyped, updated, and published with GitHub Pages - https://pages.github.com/
    - Commits can be pushed individually to the website repository for automatic publishing
- Standards and Best Practices
    - HTML – Guidelines seen here
    - JS – Guidelines seen here
    - CSS – Guidelines seen here
    - SQL – Guidelines seen here
    - Naming convention: Camel Case (Ex: firstName, lastName)

- Assumptions
    - People will have access to internet
    - There is an over-availability of electrical devices and related items (cables/wires/adapters/etc.)
    - People will have a need for spare parts
    - Owners are willing to offer surplus electronics
    - People will have access to or the ability to transport themselves to other areas
- Constraints
    - Parts will only ever sell if people are open to selling them
    - Website relies solely on user interaction
    - Users see limited listings on a location basis
    - People will need to figure out delivery of the products on their own
    - Residents may go to other sites to sell their devices instead

- Design Overview
    - Architectural Style (K):
        - GitHub Pages: GitHub Pages is a web hosting service provided by GitHub that allows users to publish static websites directly from their GitHub repositories. By creating a separate branch or directory within a repository and adding HTML, CSS, and JavaScript files, users can build a website that is automatically rendered and served as a live web page. GitHub Pages can be leveraged for collaborative design by allowing multiple contributors to work on different branches or forks of a repository, each making changes to the website's design or content. Through pull requests and version control, collaborators can review, discuss, and merge their changes, facilitating seamless collaboration and design iteration. This enables a distributed team, like us, to work together, track our modifications, and showcase our design progress in real-time.
        - The website will reflect common reselling websites like Facebook Marketplace, Craigslist, Ebay, etc.
    - Technologies being employed:
        - HTML based website
        - Integration with Google maps
        - SQL Database management for listings, forum posts, and component specs
    - Testing methodology (K):
        - Unit testing:

        - A primary component of testing for each specific program on our application. These will each be created and tested to ensure that their specific requirements are met upon their creation. E.G, creating the program for constructing a listing will be tested by verifying a listing can be created with all its respective content, and listing data will make it safely onto the database for other users to see.
    - System and Integration testing:
        - This will follow subsequent unit tests are completed to ensure all the individual program components work and operate as expected with each other. E.G, users can create an account, make a listing, respond to another listing, and post on the forum all seamlessly without interruption or data loss and any other miscellaneous bugs. If tests fail, the integration between the two components would be further analyzed to ensure there are no faults in their unit testing, before debugging any bugs that may occur upon their integration.
- General Items:
    - Homepage
    - Listing navigation system
    - Search bar system
    - Search terms
    - Account settings
    - Login/out prompt
    - Forum homepage
    - Forum navigation system
    - Forum posting system
    - Forum replying to system
    - Seller rating
- Diagram:



Application Architecture:

- List of programs:
  - Create Listing
  - Respond to Listing
  - Account Set up
  - Edit Account info
  - Search/filter function
  - Leave Review
  - Rate seller
  - Rate forum post
  - Forum Post
  - Forum response
  - Message user

List of classes:

Class: Post
+Owner +Content +Comments +Ratings
Comment() Rating()
Class: Listing

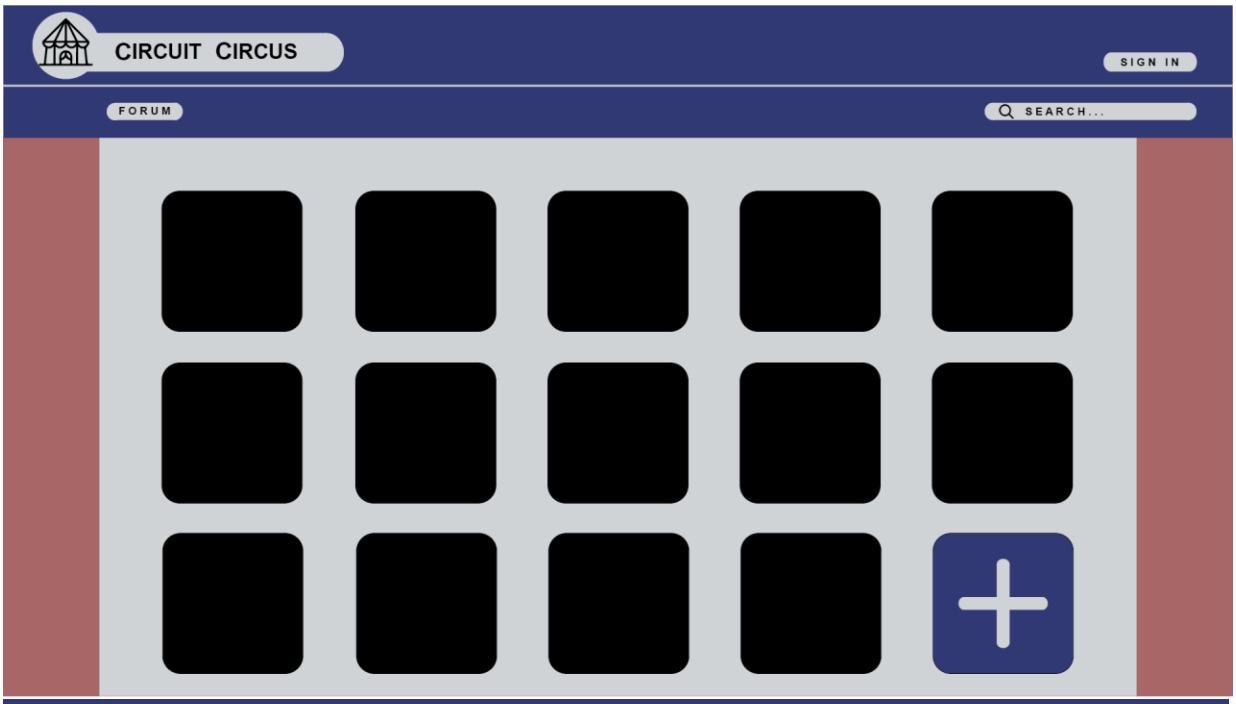
+Picture +Description +Owner +Name +Date +Condition
Respond() Edit()

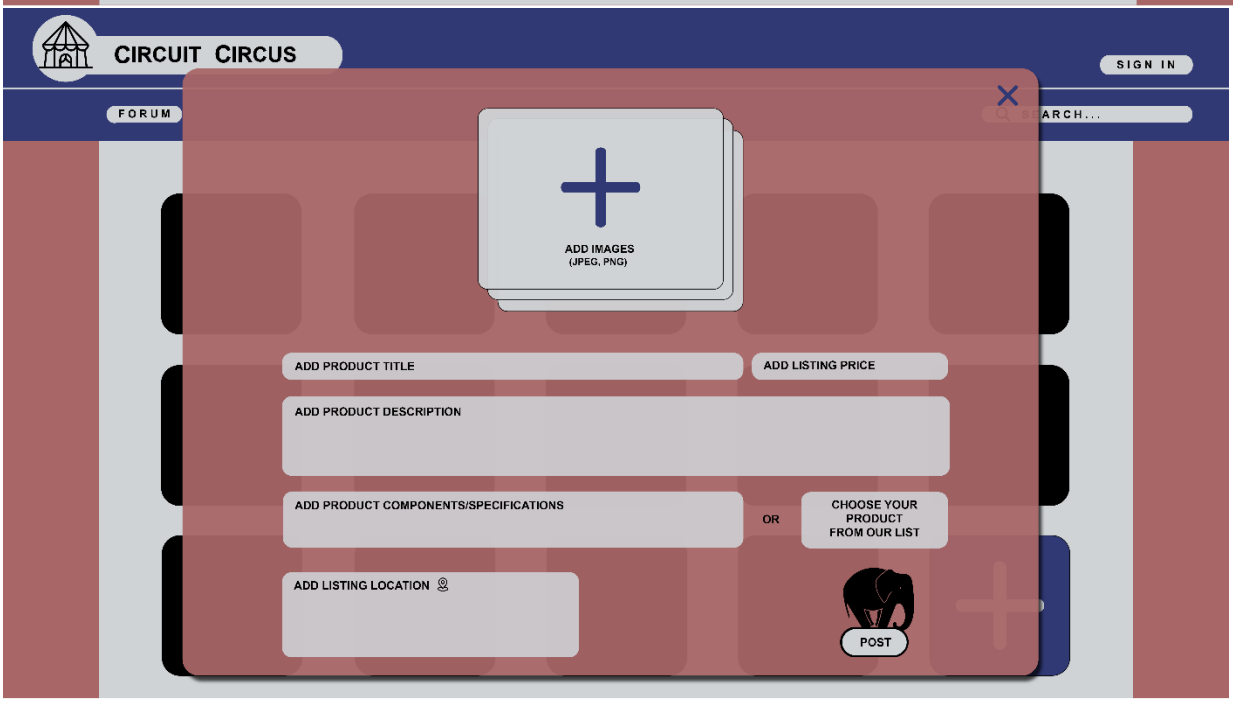
Class: User
+Username -Password +Location +Email +Review -Recs +Name +Bio +Picture
EditX() leaveReview() createAccount()

List of data types:

- Conditions for listing: I.e New, like new, good, fair, weathered, poor, shambles.
- Status: Open, Pending, Closed

GUI screens:





Access Matrix

	Products Listed	Messages	Forum
Guests	Browse	None	Browse
Buyers	Browse, Message about purchasing	Interact with their own	Browse, interact with other posts, interact with/edit their own posts, Remove their own posts
Sellers	Browse, Edit their own listings	Interact with their own	Browse, interact with other posts, interact with/edit their own posts, Remove their own posts
Administrator	All	All	All

Database Architecture:

List of tables:

Users table: Stores information about registered users, such as username, email, password, contact details, and other relevant user details.

Listings table: Stores the details of individual listings created by users, including information like item description, price, location, category, posting date, wear, and seller/user information.

Filters/Categories: Stores the possible keywords and categorizations of certain items that may appear on the platform like monitors, CPUs, motherboards, cables, modems, etc.

Forum table: The forums messaging functionality between users may include a table to store messages with attributes like sender, receiver, content, timestamp, etc.

List of views:

Popular Items View: A view that combines data from the listings table to display popular or trending items based on factors like views, ratings, proximity, or recent activity.

Search Results View: A view that combines data from the listing and categories keyword tables to provide search results based on user queries or category filters.

User's Recommendation View: A view that displays all the posts created by a specific user, combining data from the Users and Search history tables.

List of stored procedures:

Insert Listing Stored Procedure: A stored procedure that handles the insertion of new posts into the Listings table, ensuring data integrity and performing necessary validations.

Update Listing Stored Procedure: A stored procedure that allows users to update their existing posts, perform necessary checks and update the relevant data in the Listings table.

Delete Listing Stored Procedure: A stored procedure that handles the deletion of posts from the Posts table, ensuring cascading deletes and maintaining data consistency.

These would be similarly implemented for messages on the forum

Create Response: This stored procedure would handle the creation of responses to listings. It would take parameters such as the listing ID, user ID, message content, and contact information. The procedure would validate the inputs and insert the response into a responses table, associating it with the corresponding listing.

Get Listings: To retrieve listings based on various criteria (e.g., category, location, keyword), a stored procedure can be used. It would accept parameters specifying the search criteria and return the relevant listings as a result set.

User Authentication: A stored procedure can handle user authentication, accepting parameters like username and password. It would validate the credentials against the user's stored information and return a result indicating whether the login was successful or not.

List of triggers:

New User Registration Trigger: A trigger that automatically performs certain actions, such as sending a welcome email or assigning default settings, when a new user is registered and added to the Users table.

Post Expiration Trigger: A trigger that monitors the posting date of the items and performs actions like marking the post as expired or sending notifications when a certain duration has passed.

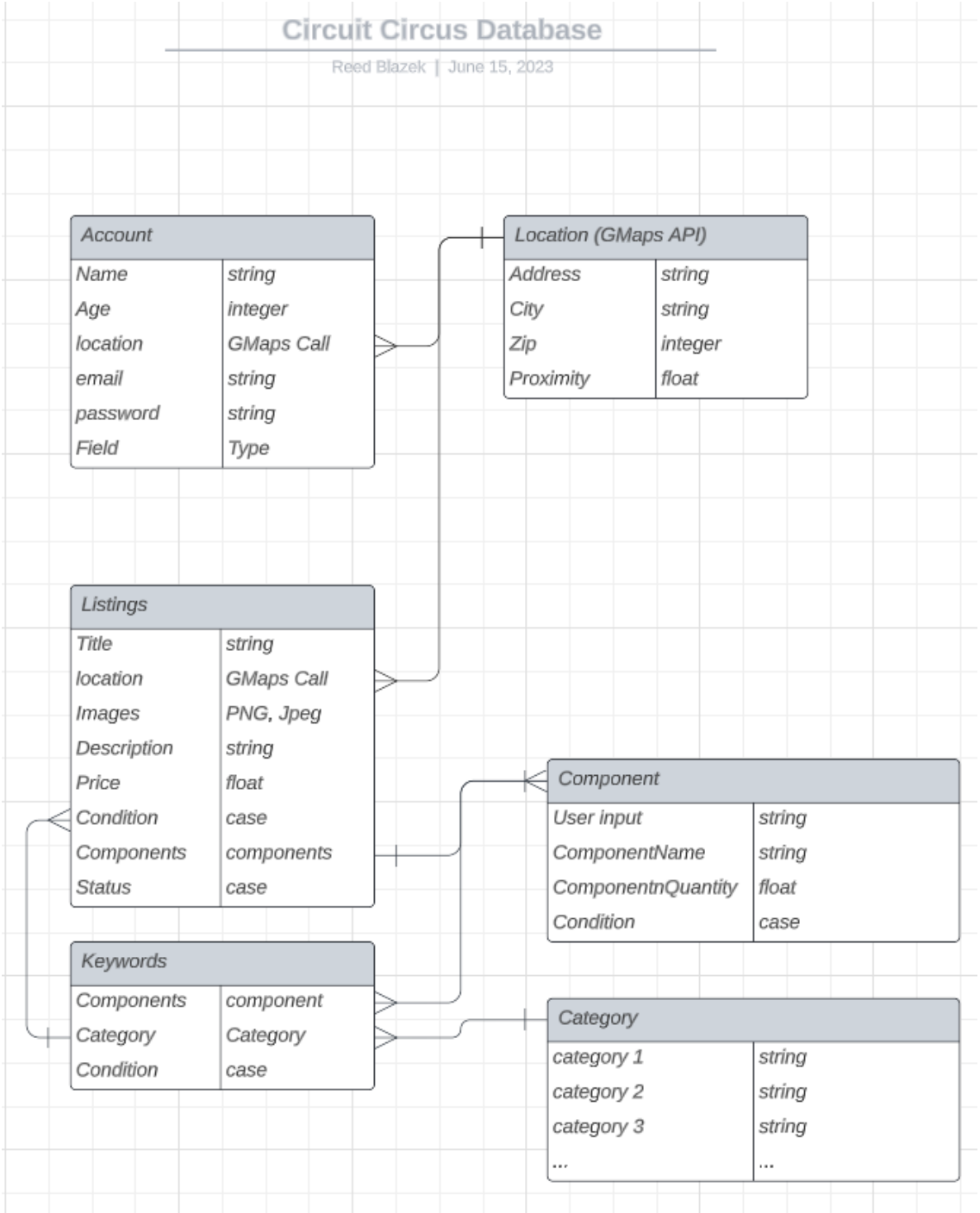
Clear listing trigger: Automatically takes down post and archives it after user marks the listing as sold.

List of jobs:

Data Backup Job: A job scheduled to run periodically to create backups of the database, ensuring data integrity and disaster recovery.

Data Cleanup Job: A job that performs regular database maintenance tasks like removing expired posts, optimizing database performance, and managing storage space.

Diagram (ERD)



Network Architecture

- Connection to high upload and download internet provided directly to our memory storage.
    - Using provided server storage of memory data sorted in RAID 10 to increase the capabilities of saving user account information, available items for sale/donation, & previously sold/donated items.
- Dedicated memory to running continuous in application usage of Google Maps for the user's location data.
- Accessible webpage will be built and hosted on GitHub Pages.

Appendices

Terms:

Listing: A listing is a class that can be uploaded to the platform by any user that displays various information and content about an item or items they possess they would like to sell/donate to other users on the platform.

Post: A post is simply a string message to be posted to the forum inquiring about projects or items people may be interested in.

Review: This is a review others can make after making a deal with another user on the platform and can only be left after both parties' consent and verify they have indeed made a deal. This becomes a way to build responsibility and trust between users and fosters a healthy community on the platform.

Rating: These are ratings that can be left on posts or listings discussing the content of said posts or listings. These are intended to boost the popularity of desirable posts and users.

Data Dictionary:

Listing Details:

Data	Data Type	Required?	Control	Description
Photo(s)	jpeg	N	User	Photos of the item a Seller would like to include
Name	string	Y	User	The name of the item(s) that are being offered
Date	String/int	N	App	The date of the post was created
Condition	Custom	Y	User	This will be selected from a pregiven list of option to the User

Description	String	Y	User	This will be a concise description of the item(s) being offered written by the user to provide more details that are not already covered by the details
Owner	Custom	Y	App	This would be the owner of whoever made the post

Account Details:

Data	Data Type	Required?	Control	Description	Example
Username	String	Y	User	The username for account, will be used to login and could be displayed to other users	Weeniebutt123
Password	String	Y	User	The password for the user's account, this will be private and used to login to the user's profile	ArcticMonkey78
Location	String?	Y	User/App	This can either be manually entered by the user if they reject location services being used, or can be pulled directly from google maps API	Cincinnati, OH
Email	String	Y	User	The Users email can be used to login in with, to verify the account, to change passwords, and to receive notifications	Example@gmail.com
Reviews	Custom	N	App	These will be a collection of reviews from other users about a user based on their merit and quality after interactions on the app have been made	"4.8/5.0 Click to see reviews"
Recommendations	Vector<posts>	N	App	These will be posts collected via an algorithm on the application and be a collection of posts that are relevant to the users search history that will be more likely to appear on the front of their browse page	"Recommended: RTX 3060, RTX 1660, RTX 3050"

Name	String	N	User	The User's given name could be collected and displayed to be more intrapersonal with people on the platform.	Dan
Bio	String	N	User	This will be a short-written portion of a user's profile that can be seen by other people to get a sense of what they are interested in or may have to sell	"Hello, my name is dan and I work at a local school doing IT work and have a lot of supplies left over from the school year! Look! Personally, I am looking for extra hard drives for my students!"
Picture	Jpeg	N	User	This will be an image displayed by the user's profile for personalization and Identification.	N/a

Sequence diagram