Members:
Alice Han (hanalice@usc.edu)
Oscar Hong (othong@usc.edu)
Kyler Saiki (ktsaiki@usc.edu)
Elaine Toh (ejtoh@usc.edu)
Lex Yu (lsyu@usc.edu)
Jonathon Wong (jzwong@usc.edu)

Flushr

*This is the sh*t*

Flushr is a web application, functioning as a map and having similar mechanics to Yelp. The application will map out all of the bathrooms on the USC campus, while also allowing users to rate them; These ratings can vary from cleanliness, wait time, utilities, etc. In order to leave behind a rating for a bathroom, the user must login to leave a verified review.

This project originated from the idea that many students, especially those with Irritable Bowel Syndrome (IBS), struggle with finding bathrooms; or for more particular students, want bathrooms that meet a specific cleanliness level. With Flushr, this would alleviate a lot of these troubles, as students could find where bathrooms are as well as see other ratings for a particular bathroom.

While Flushr can be used with or without an account, users with an account will be granted special access to more in-depth ratings, including specific categories such as bathroom accessibility, cleanliness of facilities, wait times, and more. These users will also be able to keep track of their "favorite" bathrooms and compare bathroom ratings with each other. Without an account, users can still receive the core functionality of Flushr by being able to see bathroom locations and the overall ratings of bathrooms.

Alice Han (hanalice@usc.edu)
Oscar Hong (othong@usc.edu)
Kyler Saiki (ktsaiki@usc.edu)
Elaine Toh (ejtoh@usc.edu)
Lex Yu (lsyu@usc.edu)
Jonathon Wong (jzwong@usc.edu)

Flushr: Technical Specification

Purpose

The purpose of Flushr is to make it easy for people to find bathrooms near them. The features in Flushr will allow users to find bathrooms based on proximity, different rankings (rating, wait time, etc.), and as well as leave ratings on these bathrooms for others to see. For the purpose of this application, we will only be implementing public bathrooms on USC's campus (including the USC Village).

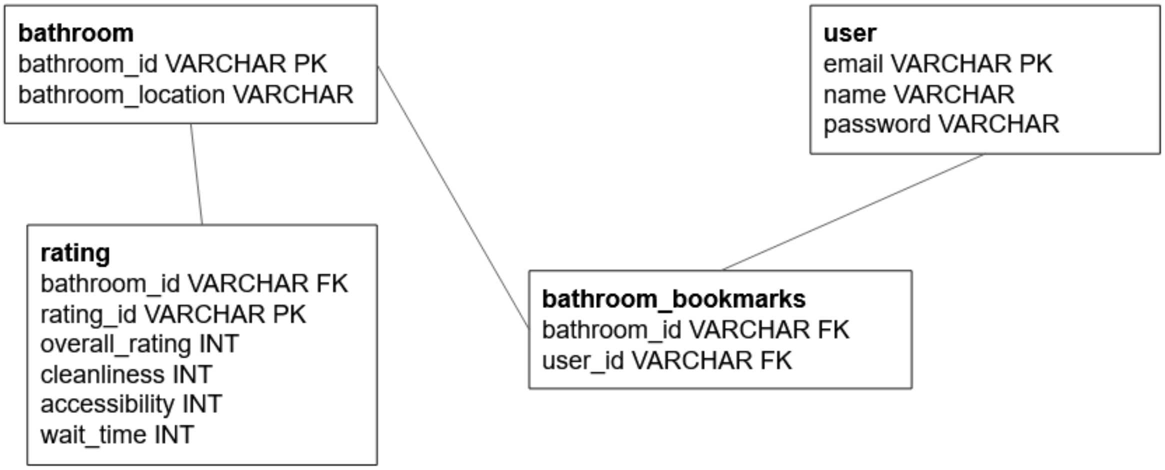
Functionality

Flushr will include a map showing the locations of bathrooms in the user's area. Users will be able to search for bathrooms, with options to sort the results by proximity, cleanliness, availability, and accessibility. Users will be able to view the information for each bathroom (provided through user ratings). Logged in users will be able to leave ratings on the bathroom's cleanliness, availability, and accessibility.

Mock-up

The mock-up on Figma is linked **here**.

SQL Diagram



bathroom
bathroom_id VARCHAR PK
bathroom_location VARCHAR

user
email VARCHAR PK
name VARCHAR
password VARCHAR

rating
bathroom_id VARCHAR FK
rating_id VARCHAR PK
overall_rating INT
cleanliness INT
accessibility INT
wait_time INT

bathroom_bookmarks
bathroom_id VARCHAR FK
user_id VARCHAR FK

Features

1. Landing page: allows the user to either sign in, register, or view the site as a guest.
    a. New user sign-up: name, email, password
    b. Returning user log-in: email, password
    c. Only authenticated users will be able to perform the following:
        i. Leave ratings on bathrooms detail page
        ii. Bookmarks page for favorite bathrooms
2. Login page: allows the registered user to log in to Flushr.
3. Register page: allows the user to register an account in Flushr.
4. Search page: allows the user to search for a nearby bathroom.
    a. Sort bathrooms by:
        i. Highest/lowest rating - provided through user ratings
        ii. Proximity
        iii. Cleanliness - provided through user ratings
        iv. Availability (Wait Times) - provided through user ratings
        v. Accessibility - provided through user ratings
    b. Filter by:
        i. Ratings (4 and above)
        ii. Proximity (within 0.1 mile)
        iii. Cleanliness (4 and above)
        iv. Availability (less than 5 minutes)
        v. Accessibility (4 and above)
    c. Bathroom details page displaying the bathroom location name and all of the ratings for the bathroom
5. Map page:
    a. Map (displaying bathrooms at their locations)
6. Bathroom page: displays each bathroom & ratings
7. Submit ratings page: allows a logged in user to submit a rating.
8. User profile:
    a. Log-out functionality
9. Bookmarked bathrooms:
    a. Shows all of the user's bookmarked bathrooms

Further Considerations

Flushr: Full Design

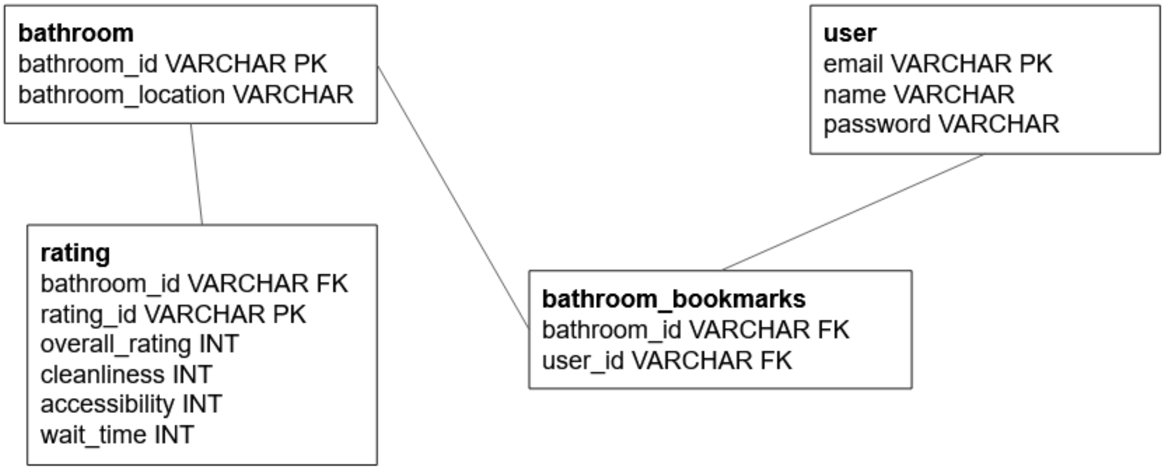
Members: Alice Han, Oscar Hong, Kyler Saiki, Elaine Toh, Jonathan Wong, Lex Yu

Hardware/Software Requirements

HTML - Front End
CSS - Front End
Javascript - Front End
Java - Backend
MySQL/MySQL Workbench - Database
Figma - Front End Mockup
GitHub/Git - Version Control
Bootstrap - CSS Framework
Flexbox - CSS Framework
AWS S3: Data Storage
AWS Elastic Beanstalk: Hosting Server
AWS Route 53: Domain Name
Google-Maps API (Java Client for Google Maps Services)

Exact GUI: The mock-up on Figma is linked **here**.

Database Schema



Difficult Algorithms/Data Structures

Landing Page:

- Nothing – we won't be using a hash table for passwords

Main Page:

- Arraylist
- Sorting (using SQL)

Feed Page:

- Arraylist

Ratings Page:

- ArrayList/Array

Profile Page:

- Nothing

Saved Page

- ArrayList

Class Diagrams/Inheritance Hierarchies

User

	User Properties
Inherits:	N/A
Public:	getters (all properties) setters (all properties)
Protected:	
Private:	String email String name String password

Rating

	Rating Properties
Inherits:	N/A
Public:	getters and setters (all properties)
Protected:	
Private:	int stars String review int cleanliness int accessibility int wait_time

Bathroom

	Bathroom Properties
Inherits:	N/A

Public:	getters and setters
Protected:	
Private:	String bathroomLocation

Flushr Testing Plan

Members: Alice Han, Oscar Hong, Kyler Saiki, Elaine Toh, Jonathan Wong, Lex Yu

White Box Unit Testing: (Test with knowledge of code):

Scenario 1: Guest User Login - Empty Fields

Test Step	Input	Expected Output
Navigate to http	Project URL	Landing Page Displayed
Click on guest login. Hit submit.	None	Page refreshes without logging in.
Click on guest login. Input an email and an incorrect password	Email and incorrect password	Page refreshes without logging in
Click on guest login. Input a correct email and password	Email and correct password	Main Page Displayed and user is logged in

Scenario 2: Searching for Bathrooms - Different Search Criteria

Test Step	Input	Expected Output
Navigate to Main page	Click on the main page button	Main page displayed (the page that lets the user search)
Select an option to sort bathrooms, then click search	Select a category to sort bathrooms by: Rating, Proximity, Cleanliness, Availability, Accessibility. Click the search button. (Repeat with all the categories)	The search results are displayed on the page, sorted by their scores in the selected category.
Select options to sort and filter bathrooms, then click search	Select a category to sort bathrooms by: Rating, Proximity, Cleanliness, Availability, or Accessibility. Select a filter: Ratings (4+), Proximity (within 0.1mi), Cleanliness (4+), Availability (<5min), or Accessibility (4+). (Repeat with all the categories and filters)	The search results satisfying the filter are displayed on the page, sorted by their scores in the selected category.
Click search	Click search without clicking any categories or filters	No search results are displayed

After searching, click on a bathroom's result.	Click on the result.	The bathroom details page for the selected bathroom is displayed and it shows the bathroom's ratings and an image of the bathroom.

Scenario 3: Leaving a rating

Test Step	Input	Expected Output
Navigate to a bathroom's details page	Click on a bathroom's result after searching.	The bathroom's details page is displayed
The user is not logged in	None	There is no option to leave a rating
The user is logged in	None	There is an option to leave a rating
Enter the rating information in the form and click submit.	Enter a string in the comments box, and select the appropriate values for the other fields.	The rating is submitted and the page is updated

Black Box Testing: (Test without knowledge of code)

Ask a non-developer to use the web application and test all the features. Record any unexpected output or unhandled cases.

Scenario 1: Unexpected functionality

Test Step	Input	Expected Output
We give the user the web application and ask them to try using it.	N/A	The user is able to successfully perform all the actions described in the White Box testing. An unexpected output would be if they found an edge case not covered in the previous testing.

Regression Testing: (test if it functions as expected after code changes, updates, or improvements)

After making code changes as prescribed by the previous tests, perform the white box and black box tests again.

Deployment Steps

Opening Eclipse Webapp
1 - Download and Open Zip File
2 - Go Into Eclipse -> File -> Open Projects from File System
3 - Select CSCI201_FinalProject_Flushr

Setting up SQL
4 - Make Sure MySQL Workbench is installed at version
5 - Create a new MySQL Connection by clicking the Plus Sign and changing the Username to root and password to root1234
6 - Open the Connection
7 - File -> Open SQL Script -> Flushr -> sql -> Flushr_DB.sql -> Open
8 - Click the Lightning Bolt Icon (Run the Script)
9 - Click the Refresh Button

Setting up Tomcat Server (Skip if a server is already setup)
10 - Open Eclipse -> Window -> Show View -> Servers
11 - Create a New Server and set it to Tomcat v9
12 - Start Server
13 - Run the Eclipse Application and run it on that server

Finished

Kyler Saiki
Elaine Toh
Lex Yu
Jonathan Wong
Alice Han
Oscar Hong

Video Link

Link:
https://drive.google.com/drive/folders/1uO49pLLKzOUc_bRTUH5QuXbetbLBhNqL?usp=sharing

FLUSHR:

this is the sh*t.

CSCI 201 Final Project, Spring 2022

—

Alice Han, Oscar Hong, Kyler Saiki, Elaine Toh, Jon Wong, Lex Yu

Designed by USC students…

…for the IBS community at USC.

TABLE OF CONTENTS

01. IDEATION & DESIGN

Initial project design, teamwork dynamic, etc.

02. PRODUCT

Introduction of the web application *FLUSHR*.

03. DEVELOPMENT

Discussion of key features, requirements met, and outside curriculum applications.

04. DEMO

Formal product demonsration

01.

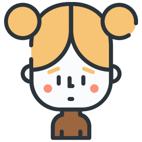
IDEATION & DESIGN

Initial project design,
teamwork dynamic, etc.

ROLE DELEGATION

KYLER SAIKI

Team lead
Full-stack developer



LEX YU

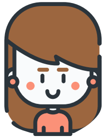
Backend developer

ELAINE TOH

Backend developer

OSCAR HONG

Frontend developer

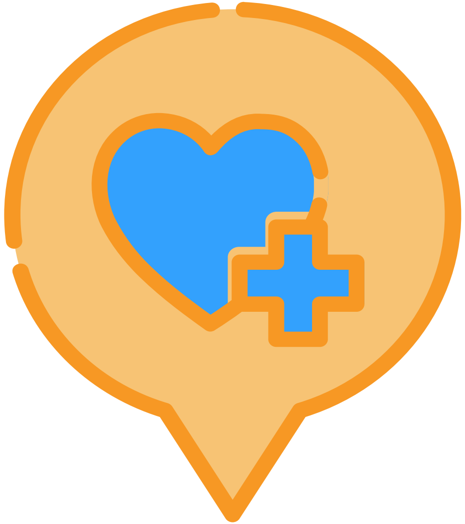


ALICE HAN

Designer
Frontend developer

JON WONG

Field researcher
Analyst



IDEATION

Flushr first gained inspiration from the IBS (*irritable bowel syndrome*) community, our intended end users. IBS is a common, chronic GI (*gastrointestinal*) disorder affecting the digestive system — this often means they need to be prepared to find a bathroom near them at all times.

Flushr makes that easy.

Although the IBS community is our intended audience, this can be used universally by all USC students. Whether you're in a hurry or unfamiliar with the campus, Flushr can sure be handy in the right time.

CREATION

The purpose of Flushr is to **make it easy for people to find bathrooms near them**. The features in Flushr will allow users to find bathrooms based on proximity, different rankings (rating, wait time, etc.).

For the purpose of this application, we will only be implementing public bathrooms on **USC's campus** (including the USC Village).

DESIGN

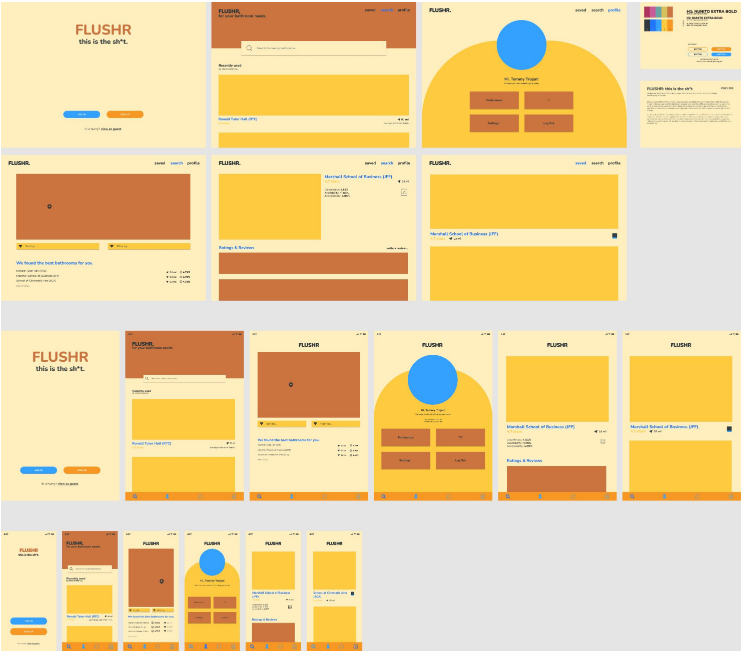
Flushr will include a **map** showing the locations of bathrooms in the user's area. Users will be able to search for bathrooms, with options to sort the results by **proximity**, **cleanliness**, **availability**, and **accessibility**. Users will be able to view the information for each bathroom. Logged in users will be able to view a profile page.

02.

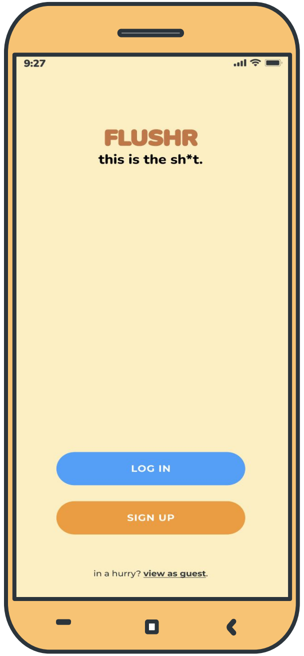
PRODUCT

Introduction of the web application *FLUSHR*.

GUI MOCKUP + PROTOTYPE



RESPONSIVE WEB APPLICATION MOCKUP

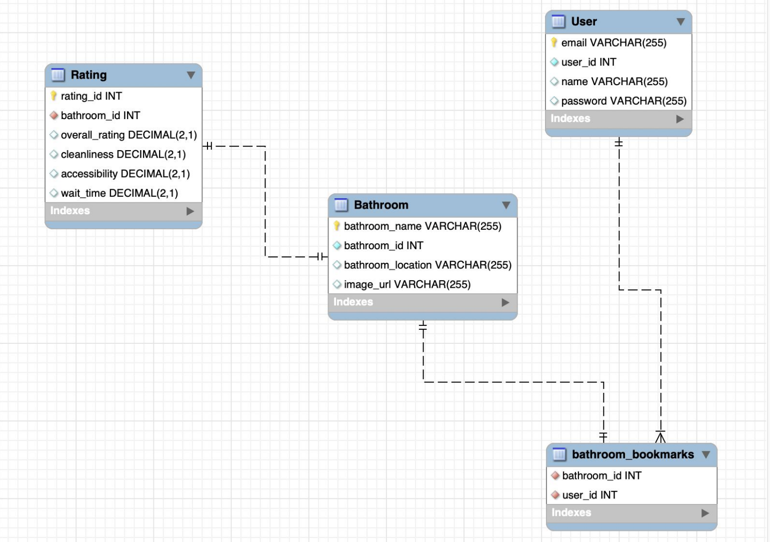


MOBILE PROTOTYPE

03.

DEVELOPMENT

Discussion of key features, requirements met, and outside curriculum applications.



User

email VARCHAR(255)
user_id INT
name VARCHAR(255)
password VARCHAR(255)
Indexes

Rating

rating_id INT
bathroom_id INT
overall_rating DECIMAL(2,1)
cleanliness DECIMAL(2,1)
accessibility DECIMAL(2,1)
wait_time DECIMAL(2,1)
Indexes

Bathroom

bathroom_name VARCHAR(255)
bathroom_id INT
bathroom_location VARCHAR(255)
image_url VARCHAR(255)
Indexes

bathroom_bookmarks

bathroom_id INT
user_id INT
Indexes

DATABASE DESIGN

USER LOGIN FUNCTIONALITY

REGISTERED
USER

MULTITHREADING & NETWORKING

MULTITHREADING

A simulation of a rock-paper-scissors tournament.

NETWORKING

Used in deployment

SOFTWARE AND HARDWARE USED

HTML/CSS

HTML/CSS was the main language used for building the frontend of the application.

FRONTEND

JAVASCRIPT

Javascript was also used supplemental to the limitations of HTML/CSS.

- Event handlers/listeners

FRONTEND

JAVA

Java was used to handle the backend of Flushr.

- Servlets
- Sorting algorithms

BACKEND

MYSQL

SQL was the main database framework used to connect data to the application.

- MySQL Workbench
- Internal database design

BACKEND

IMPLEMENTATIONS OUTSIDE CURRICULUM

FIGMA

Figma was used to design an initial mockup and prototype.

ELASTIC BEANSTALK

AWS was used to host the website on a server.

GIT/GITHUB

Git was used to make collaboration simple when handling code.

API

Flushr makes a call to the Google Maps API for location purposes.

BOOTSTRAP

Bootstrap, a CSS framework, was used to make responsive design.

ROUTE 53

Route 53 was used to get the domain name for the webapp.

04.

DEMO

Formal product
demonstration.

THANK YOU

Any questions?