CrapMap

CSCI 4176/5708 Term Project

Team #2

Geoffrey Caven - caven@cs.dal.ca
Stanford Lockhart - lockhart@cs.dal.ca
Brandon Min - br239881@gmail.com
Niclas Skaalum - niclas.skaalum@dal.ca
Carson Swinamer - swinamer@cs.dal.ca

Abstract

CrapMap is a native Android application for reviewing public bathrooms. The application uses the device's GPS functionality to display and interactive map of the user's location marked with public bathrooms. The user can view both high and low level bathroom information, including hours of operation and user reviews and ratings for cleanliness, availability, and accessibility. The user can either log in or stay anonymous to leave a review for a bathroom. All reviews and user information are stored in a database through an API hosted on Heroku.

GitHub repository link: https://github.com/salockhart/CSCI-4176-Project---Team-2

Completion Report

The project is in a state that reflects our original goals in most aspects including appearance, functionality, and timeline. Features were mostly implemented excluding the camera and picture functionality. Additional work was put into features that would provide polish and usability to the app.

Issues arose during the development stages of our project, as the lack of proper GPS functionality on our emulator software slowed the testing. Basic UI testing was able to be done, but back-end integration was slowed due to this. Certain members also did not have an Android phone, making testing for their work more difficult.

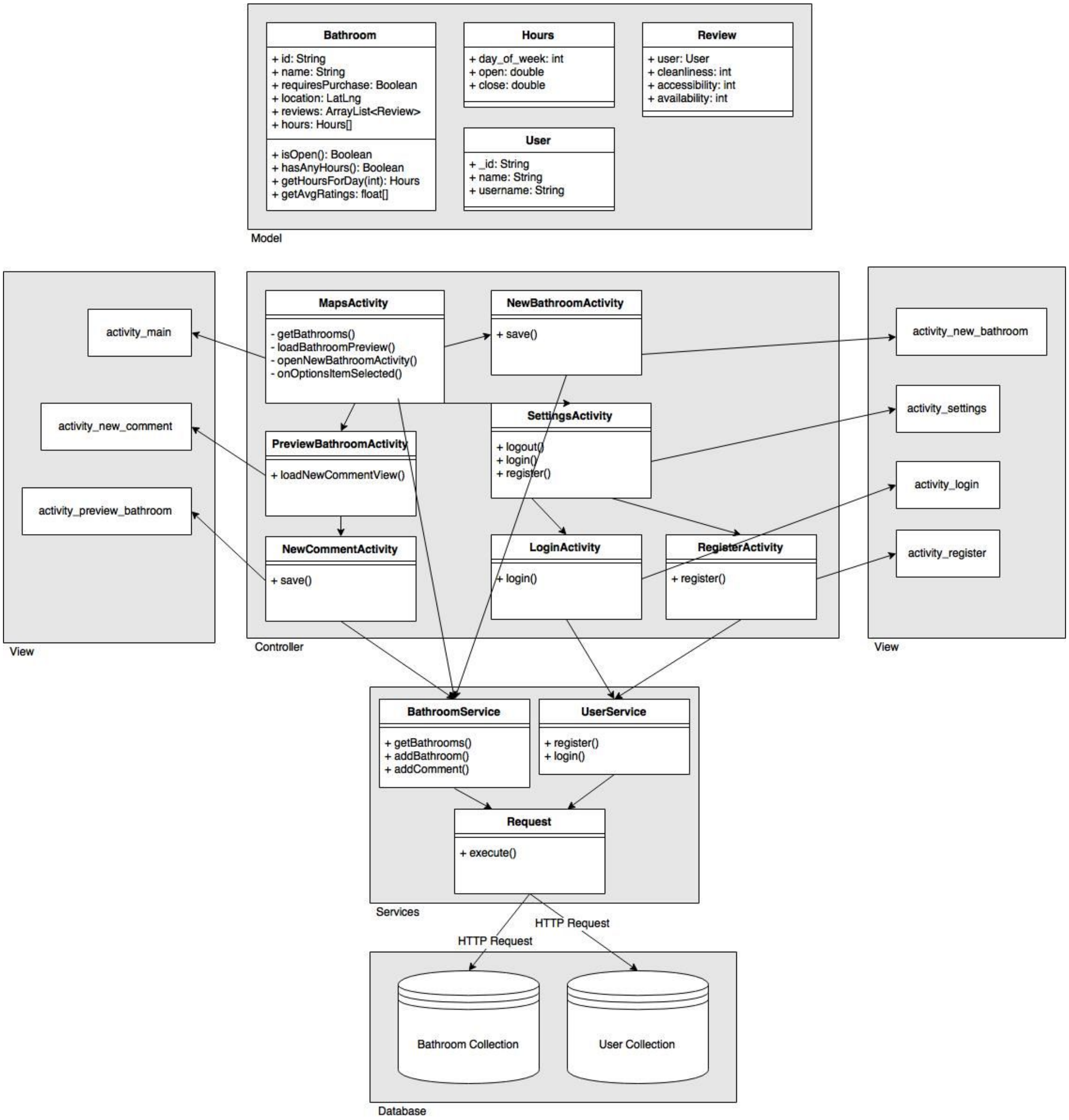
Camera functionality was deemed unnecessary due to the complexities involved with storage and transmission from our database. It was moved to a stretch goal, but progress never begun due to time constraints. Additional features were added to replace this however, with accelerometer detection for recommended bathrooms, icons for requiring purchase, and general polish to make our application more user friendly.

Below is the timeline provided in our project update and proposal documents, updated to the current state of the application. All tasks have been completed, resulting in a finished product that reflects our original goal.

Week	Task(s)	Status	Assigned to:
Feb 13 - Feb 19	1. Finalize Initial Design	Complete	All
	2. Wireframes	Complete	All
	3. Plan basic code structure	Complete	All
Feb 20 - Feb 26	4. Prepare Code Structure	Complete	All
	5. Create backend classes	Complete	All
	6. Prepare server	Complete	All
Milestone 1: Proof of Concept & Basic Structure			
Feb 27 - March 5	7. Prepare group tasks	Complete	All
	8. Finalize Wireframes	Complete	All
	9. Finalize Clickstreams	Complete	All
	10. Finalize Site Map	Complete	All

	11. Prepare for Development	Complete	All
March 6 - March 12	15. Map View	Complete	Geoff
	16. Database & back end	Complete	Stanford
	17. New Bathroom View	Complete	Niclas
	18. Detail View	Complete	Carson
	19. Comment View	Complete	Brandon
	20. Wireframes into Android	Complete	All
Milestone 2: Working Prototype			
March 13 - March 19	21. Link Map View to Backend	Complete	Geoff
	22. Link Backend to Front	Complete	All
	23. Link Bathroom to Backend	Complete	Geoff
	24. Link Detail View to Backend	Complete	Carson
	25. Link Comment View to Backend	Complete	Brandon
	26. Basic Testing	Complete	All
March 20 - March 26	24. Finalized View Structures	Complete	All
	25. All Expected Functionality	Complete	All
	26. Graphical polish	Complete	All
	27. Testing & bug fixing	Complete	All
	29. Any bonus functionality	Complete	All
	30. Accelerometer shortcuts	Complete	Geoff
	31. User accounts in app	Complete	Stanford
Milestone 3: Fully Functional Application			
March 27 - April 6	30. Continued testing	Complete	All
	31. Presentation preparation	Complete	All
	32. Final application	Complete	All
Milestone 4: Ready for Presentation			

Functional Decomposition



The application uses a Model View Controller pattern, as well as a Services pattern for interacting with the database.

The Services pattern is used to abstract away the interface to the databases. We use a backend API to interface directly with the database, and the Services layer in the application makes the HTTP requests to the API and receives the results back. This way, the app would be portable to any back end, and could be mocked if we wanted it to be.

The Model View Controller pattern is what dictates the rest of the application. For every View, there is a Controller - the Controller then handles the parts of the Model that it requires, and the Models get passed from Controller to Controller as the user navigates the application.

The Model is a set of classes that represent the information as it flows through the application. Each Model is a different data object, and they aggregate together to contain the full state of the application. For instance, the Bathroom model has within it a list of Review objects, as well as a list of Hours objects. By passing a Bathroom model, we ensure that we get all of the data associated with it.

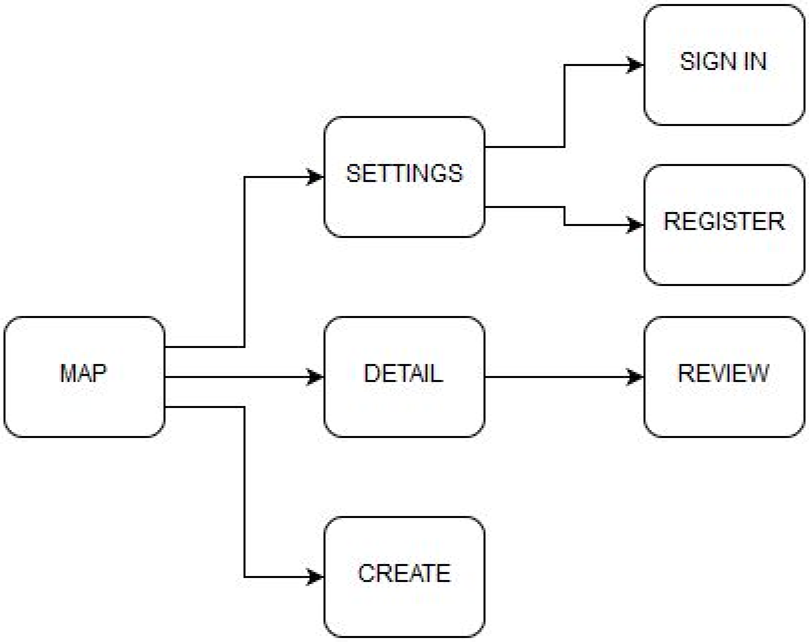
Each View is contained in a single XML file, with some extra XML files to be used in view adaptors. The Views have the buttons that call the functions specified in the diagram above. Every View has it's own Controller that handles the events for that View, and can interface with the Model as well as the Services to accomplish tasks.

The Controllers are what handle the navigation through the application, and display the data contained in the Model as part of the View it is associated with. Not every Controller has a full view of the Model, but just enough to display what it needs to.

An example of this in action is the transition from the Map View to the Preview View. The Map View, on load, will ask the Bathroom Service to give it the list of bathrooms. When it receives them, it updates its model to be a list of Bathroom objects. These objects are what it then displays in the View as pins on the map, and dialogs that appear when one of those pins are tapped. When the dialog is tapped, the Preview View is opened, and the Map View passes along the tapped Bathroom model to it. As far as the Preview View is concerned, there is only one Bathroom and the Services don't exist. It just needs to display the data in the View, and handle the events that occur when someone wants to add a new Review.

High-level Organization

The product high-level organization (shown below) remains largely unchanged since previous project updates. The Map activity serves as the default activity connecting to other activities. Selecting a bathroom pin on the map displays summary information for the bathroom, and selecting again navigates to the Detail activity. The Detail activity shows more in depth information for the selected bathroom and allows the user to submit a review via the Review activity. From the Map activity, a user can also create a new bathroom through the Create activity, or sign in or register through the Settings activity.



Clickstreams

The following are some common use cases for CrapMap:

Browse nearby bathrooms

1. User opens the app. The Map activity loads at the user's current location. Nearby bathrooms are marked with pins on the map. Currently open bathrooms are marked with green pins, while closed bathrooms are marked with red pins.
    a. If necessary, the user drags across the map to see other locations. User pinches the map to zoom in or out. User taps the current location icon in the top-right corner to return to the current location.
2. User taps a pin. The pin expands to show high-level summary information for the selected bathroom.

Sign in

1. User opens the app.
2. User taps the settings icon in the top-right corner.
3. User selects the “Sign In” button.
4. User provides username and password in the corresponding fields.
5. User selects the “Sign in” button. The user is signed in.

Register

1. User opens the app.
2. User taps the settings icon in the top-right corner.
3. User selects the “Register” button.
4. User provides name, username, and password in the corresponding fields.
5. User selects the “Register” button. The user is registered.

Review a bathroom

1. User opens the app.
2. User taps the summary information for a pin and navigates to the Detail activity.
    a. Alternatively, the user shakes the phone to directly navigate to the Detail activity for a nearby open bathroom.
3. The user taps the add button in the bottom-right corner. The user is navigated to the Review activity.
4. The user enters ratings for cleanliness, accessibility, and availability using three 5-star rating bars.
5. The user enters a comment in the comment field.
6. The user taps the complete button in the bottom-right corner. The user is navigated back to the Detail activity. The new review appears in the Detail activity and the average ratings are adjusted accordingly.

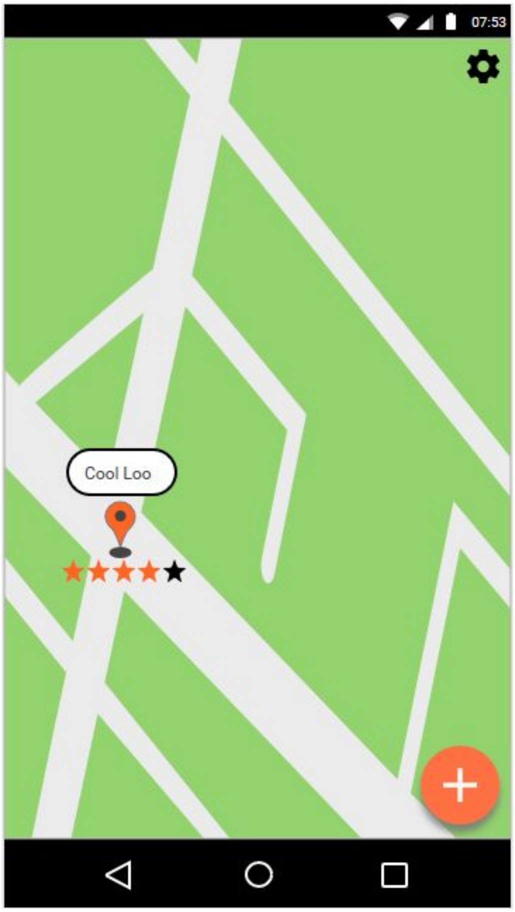
Create a new bathroom

1. The user opens the app.
2. The user taps the add button in the bottom right corner. The user is navigated to the Create activity.
3. The user enters a name for the bathroom in the corresponding field.
4. The user specifies whether the bathroom requires purchase using the corresponding checkbox.
5. The user specifies hours of operation using the corresponding checkboxes for days and time selectors for open and close times.
    a. The open and close times are populated to reflect user choices.
6. The user selects the “Submit” button at the bottom of the page.
    a. The page notifies the user if the form is incorrectly filled.
    b. If the form is correctly filled, the new bathroom is created and the user is navigated directly to the Detail view for the new bathroom.

Layout

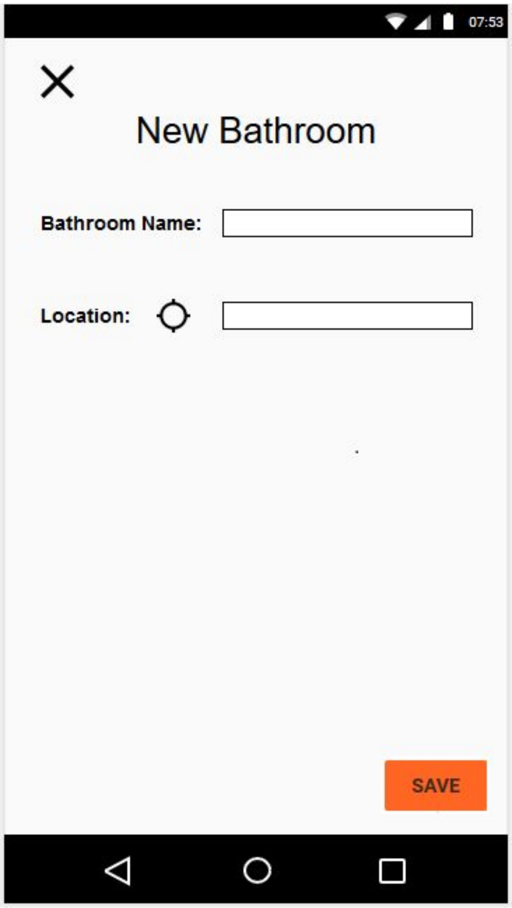
The following wireframes correspond with all the activities featured in the final product, though with some varying functionality.

Maps Activity



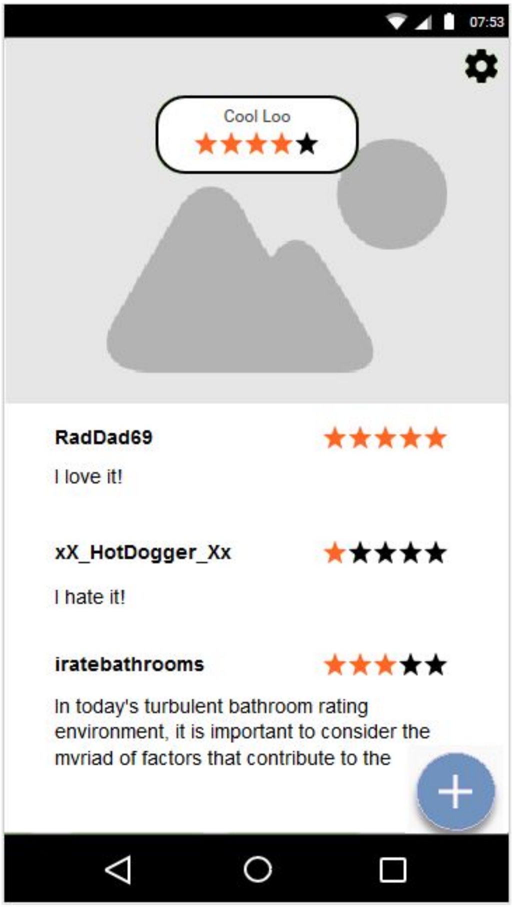
The Map activity remains largely the same as its wireframe, allowing users to view nearby bathrooms, high-level bathroom information and means of accessing other activities.

Create Bathroom Activity



The Create activity wireframe corresponds closely with its production equivalent, though the product also allows the user to specify whether the new bathroom's hours of operation and whether it requires purchase. Furthermore, the application adds new bathrooms to the current user location, rather than a user specified one.

Detail Activity



RadDad69 ★★★★★

I love it!

xX_HotDogger_Xx ★★★★★

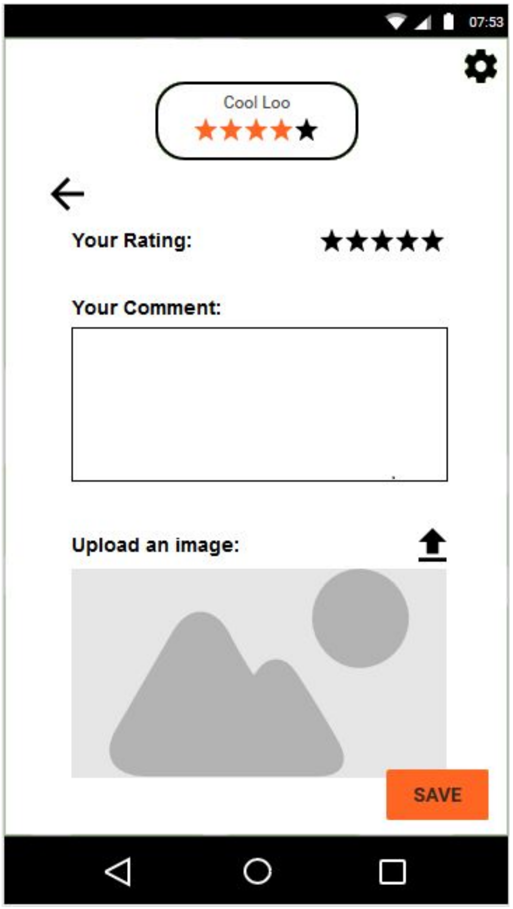
I hate it!

iratebathrooms ★★★★★

In today's turbulent bathroom rating environment, it is important to consider the myriad of factors that contribute to the

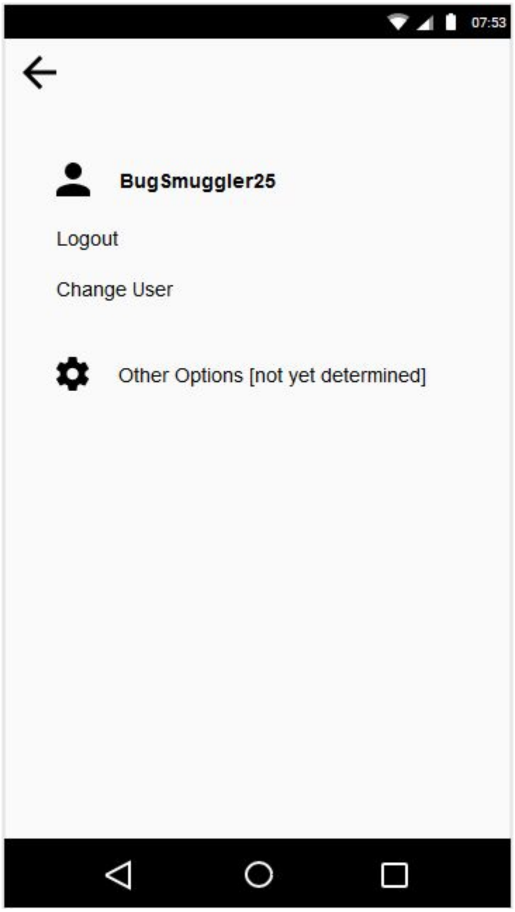
The production Details activity is very similar to its wireframe. The production activity boasts additional rating categories and a cleaner display, but does not support images as the wireframe indicates.

Review Activity



The Review activity, like the Detail activity, largely differs from its wireframe by not supporting images, but instead more rating categories. This activity allows users to rate and comment on existing bathrooms.

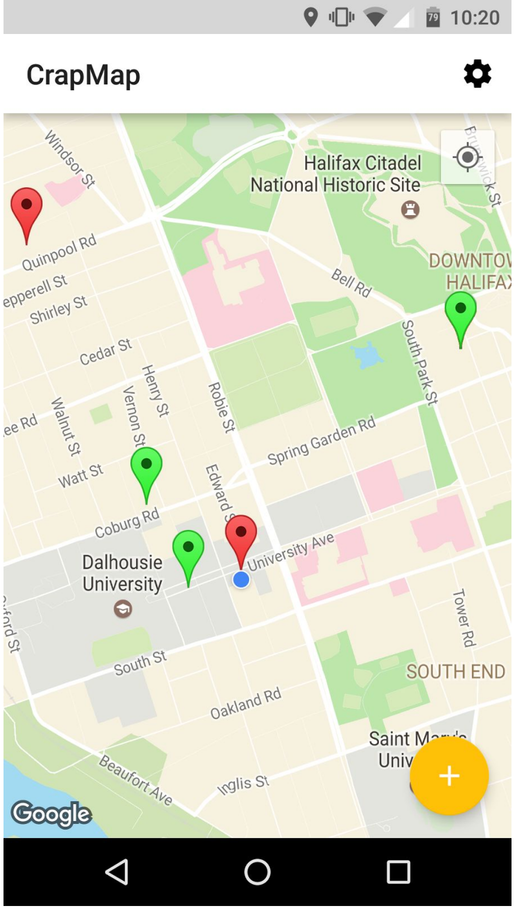
Settings Activity

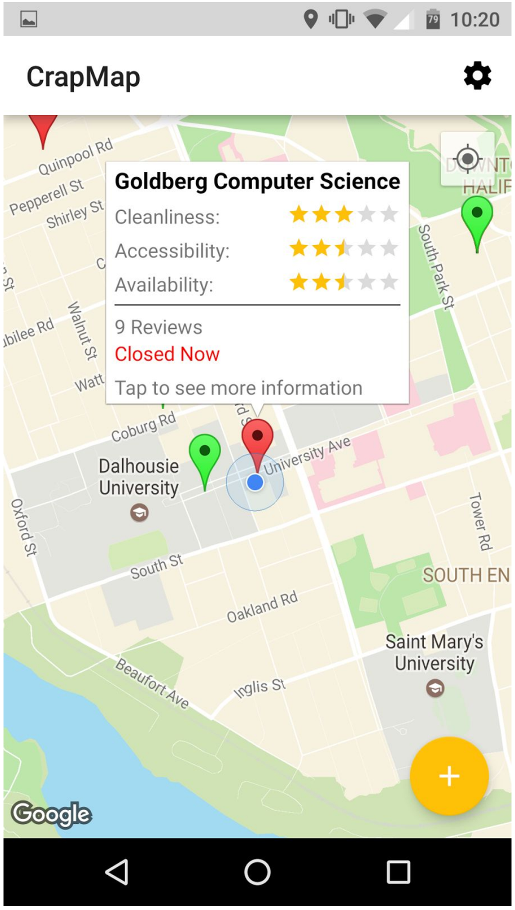


The finalized Settings activity allows its users to register and sign in, supporting the account functionality featured in its wireframe. User accounts are used to associate users with comments.

Implementation

Maps Activity





Goldberg Computer Science

Cleanliness: ★★★☆☆

Accessibility: ★★½☆☆

Availability: ★★½☆☆

9 Reviews

Closed Now

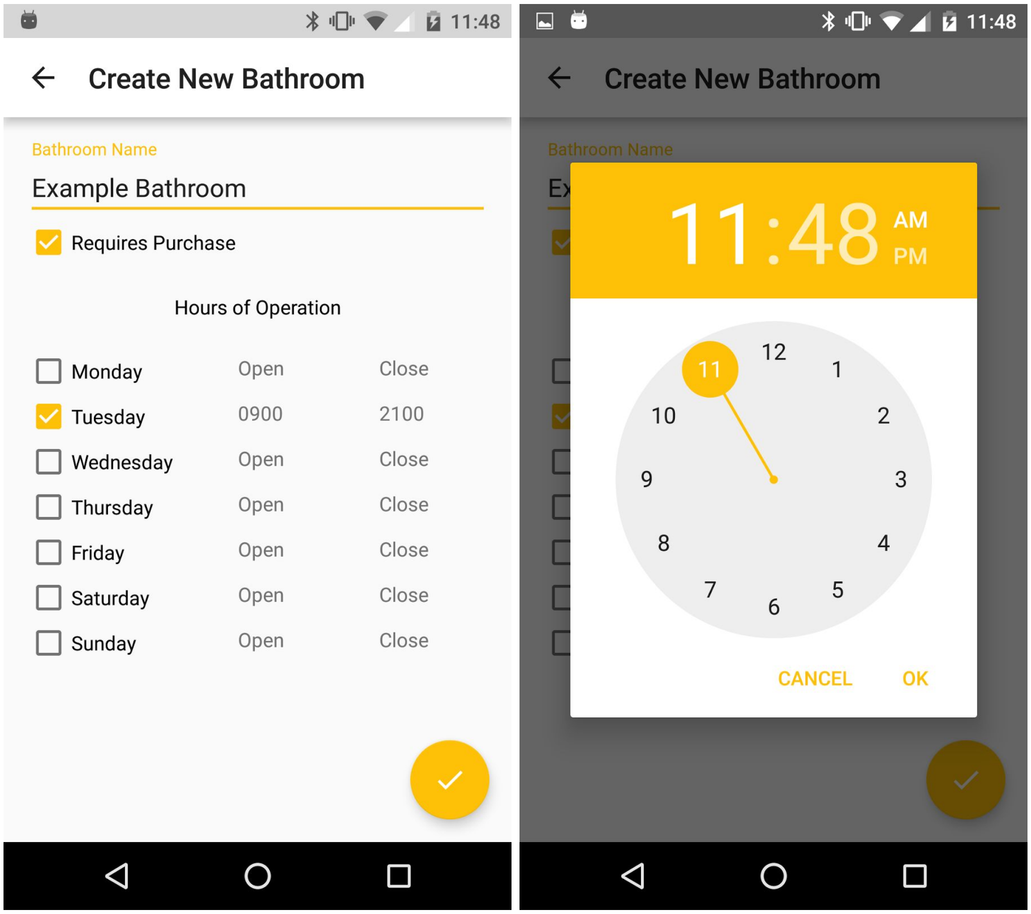
Tap to see more information

The Maps Activity has been fully implemented, with a google maps view displaying all nearby bathroom objects saved in the database. The map markers representing the bathrooms are either displayed in a colour reflected if they are currently open or not (green if the building containing the bathroom is currently open or has no hours of operation on record, red if the building is currently closed). Tapping on a marker will open a pop-up window containing a summary of information about the bathroom, including the average of reviews for each category, the number of reviews, the open/closed status, and whether that bathroom requires a purchase to access. Tapping on this window will open the Detail Activity for that bathroom.

The action bar at the top of the Maps Activity features an icon of a gear. Tapping on the gear will take the user into the Settings Activity, which allows them to create a user account, login, and logout as appropriate.

A large yellow button is placed at the bottom-right of the Maps Activity. When tapped, this will open the Create Bathroom Activity, which allows the user to create a new bathroom entry at their current location.

Create Bathroom Activity



Create New Bathroom

Bathroom Name

Example Bathroom

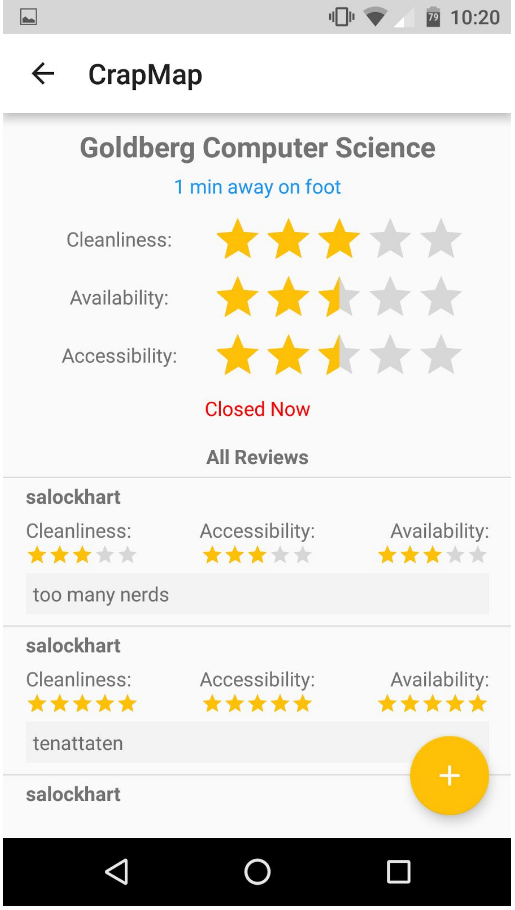
☑ Requires Purchase

Hours of Operation

☐ Monday	Open	Close
☑ Tuesday	0900	2100
☐ Wednesday	Open	Close
☐ Thursday	Open	Close
☐ Friday	Open	Close
☐ Saturday	Open	Close
☐ Sunday	Open	Close

The Create Bathroom activity allows a user to make new public bathroom entries at their current location. The user can enter a title, mark whether access to the bathroom requires a purchase, and enter the hours of operation. Tapping on ‘open’ or ‘close’ label will open a modal containing a time picker widget. Once a time has been selected, the label will be replaced with the selected time displayed in 24hr format. Tapping submit will submit the new bathroom to the database and open the Detail Activity for the new bathroom, allowing the user to immediately create a review.

Detail Activity



Goldberg Computer Science

1 min away on foot

Cleanliness: ★★★☆☆

Availability: ★★½☆☆

Accessibility: ★★½☆☆

Closed Now

All Reviews

salockhart

Cleanliness: ★★★☆☆ Accessibility: ★★★☆☆ Availability: ★★★☆☆

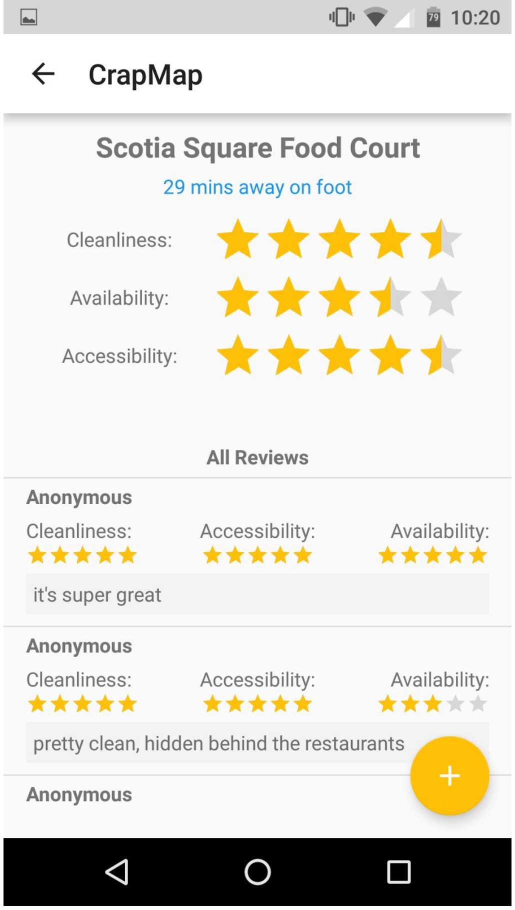
too many nerds

salockhart

Cleanliness: ★★★★★ Accessibility: ★★★★★ Availability: ★★★★★

tenattaten

salockhart



Scotia Square Food Court

29 mins away on foot

Cleanliness: ★★★★½

Availability: ★★★½☆

Accessibility: ★★★★½

All Reviews

Anonymous

Cleanliness: ★★★★★ Accessibility: ★★★★★ Availability: ★★★★★

it's super great

Anonymous

Cleanliness: ★★★★★ Accessibility: ★★★★★ Availability: ★★★☆☆

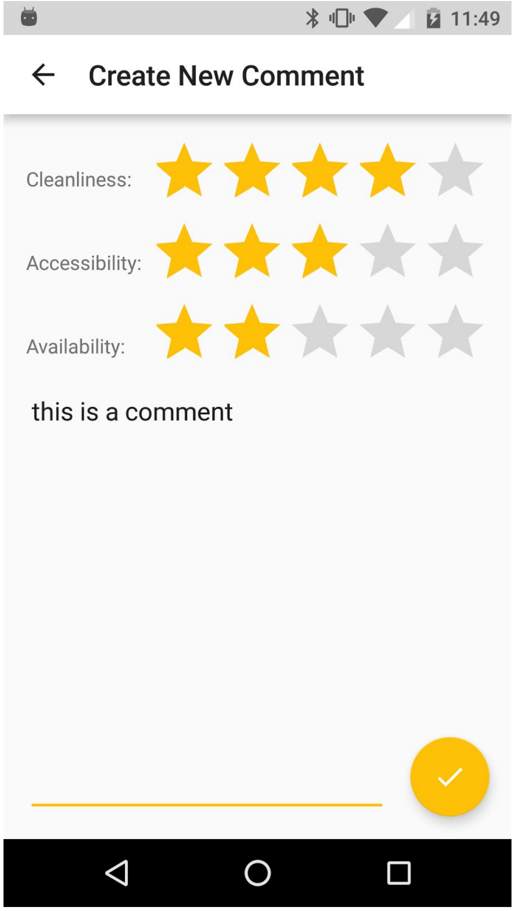
pretty clean, hidden behind the restaurants

Anonymous

The Details Activity displays a summary of the reviews for a bathroom, along with a list of all the reviews in the database. At the top of the activity is the name of the bathroom as defined by the user who created it, along with a walking time estimate from the user's current location (this is retrieved from the Google Maps Distance Matrix API in a background asynchronous request). This is followed by the same information summary seen in the pop-up on the Maps Activity. At the bottom is a complete list of all reviews on file for this bathroom. If there are more than two reviews, the entire activity can be scrolled up to reveal them.

At the bottom-right of the activity is a large yellow button. Tapping on this button will bring the user to the Review Activity, allowing them to create a new review for the bathroom whose details they are currently viewing. This button stays in place above the background content as the activity is scrolled.

Review Activity

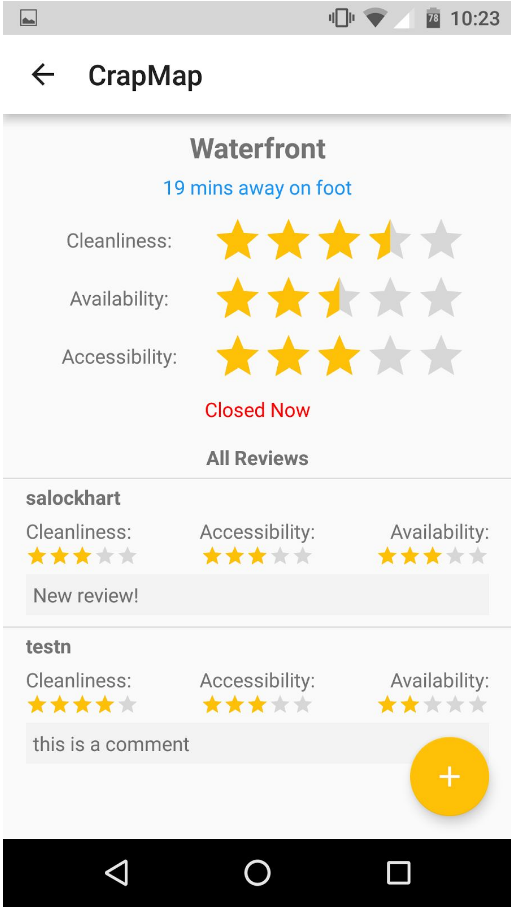


Cleanliness: ★★★★☆

Accessibility: ★★★☆☆

Availability: ★★☆☆☆

this is a comment



Waterfront

19 mins away on foot

Cleanliness:

Availability:

Accessibility:

Closed Now

All Reviews

salockhart

Cleanliness: ★★★☆☆ Accessibility: ★★★☆☆ Availability: ★★★☆☆

New review!

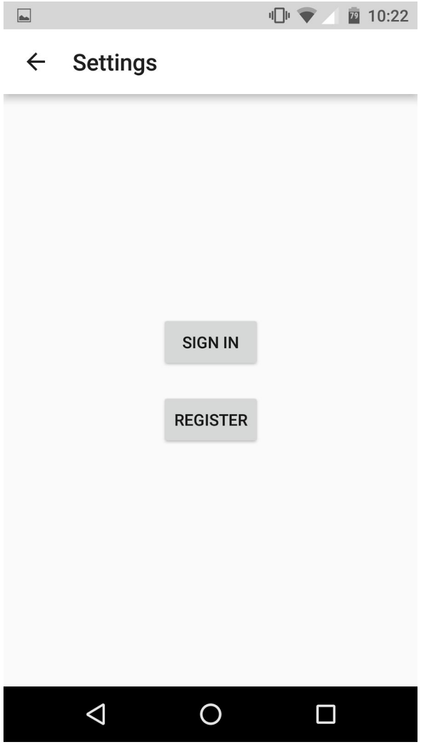
testn

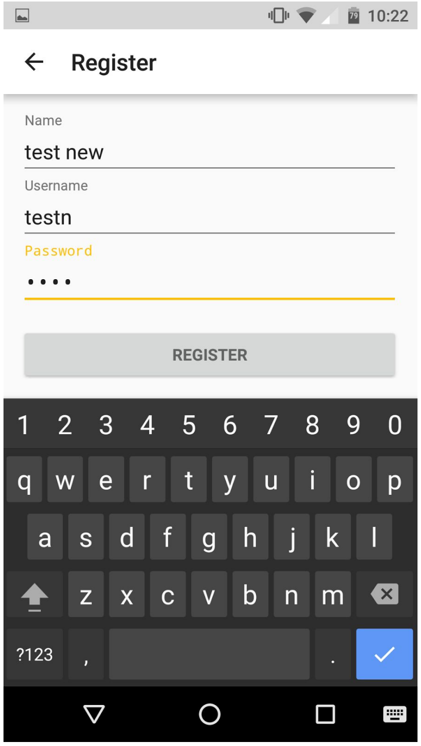
Cleanliness: ★★★★☆ Accessibility: ★★★☆☆ Availability: ★★☆☆☆

this is a comment

The Review Activity allows a user to create a new review for a bathroom. Users can tap or drag on the stars to rate the bathroom on each of the three provided metrics (Cleanliness, Accessibility, and Availability). Users can also type into the provided space to leave more detailed textual feedback. Tapping on the large yellow checkmark button will submit the review to the database and return the user to the Detail Activity, which will now include their new review. If the user was logged in when the review was created, their username will be displayed alongside the new review. Otherwise, the review will be marked as anonymous.

Settings Activity

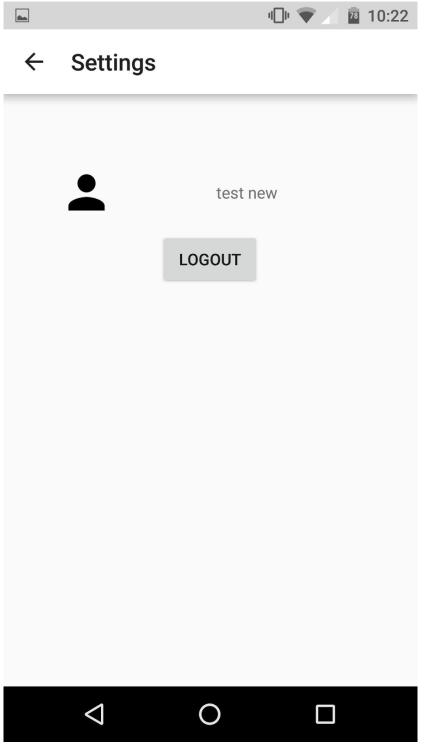




Name
test new

Username
testn

Password
••••



test new

The Settings Activity allows them to create a user account, login, and logout as appropriate. If the user is not logged-in, buttons for signing in and registering will be displayed. Tapping on either of them will bring the user to the appropriate form for logging in or registering. Registering will return the user to the original button view, allowing them to then log in. Logging in successfully will display a view showing the user's username and a button to logout. The logged-in user is stored locally and will be retained until they log out.

Future Work

There are many possible avenues for future growth in the application. The ability to upload images of a bathroom would be a good additional function, and it was originally part of the project scope. A picture would help for the user to recognise the bathroom and assess remotely on their own. The photos could also help a user to find a bathroom, since it it could be challenging for buildings that Google Maps has not mapped the interior of. Photo upload functionality would have to be judiciously moderated, however. For one, there are potential legal issues around taking photos of public bathrooms, as well as the obvious privacy concerns. User may also purposely or accidentally upload an image that could be seen as problematic to other users. A content guideline could be established, but active administration by the developer would be needed.

The bathroom review could have additional information, such as disability support and gender division. Not many public bathrooms have great disability support, so being able to locate a bathroom with better support would be helpful for disabled users. The bathroom could also be tagged as gender-neutral, so that users with that preference or for whom gendered bathrooms are problematic would be able to find accommodating locations nearby.

The settings menu of the application could be expanded to provide more functionality. It could contain an option to view the history of user reviews, and remove any of those reviews. The application does not have the ability to delete any reviews, so this would need to be implemented first. The application could also implement elements of “gamification” like counting the amount of reviews made and unlocking “achievements” or user icons to reward users for contributing, but this would be a low priority.

Finally, the application could branch out to a web application. This version would be limited in its functionality that requires GPS, but has better accessibility and would still allow viewing of any bathroom on a Google Map. It could also have a search bar to search all bathrooms based on the city. This would make viewing any bathroom far away from the current position easier, for future planning or just out of curiosity.