

Litterbug – A Software Approach to Littering

DAVID MEDNIKOV, Oregon State University

HITESH VARMA, Oregon State University

KEVIN ALLEN, Oregon State University

ALEXANDER YFRAIMOV, Oregon State University

1. REQUIREMENTS DEFINITION

Litterbug will be an app where users go out and clean up trash/litter an area. Users will earn points and badges/medals for cleaning up areas. Users will be able to challenge other users to clean up an area as well. After a user reports an area as cleaned, another user will have to confirm that the area was in fact cleaned. The points and rewards will not transfer to the first user until the second user confirms that the area was cleaned. Users will be able to donate to a charity of their choice using various payment methods including Apple/Google Pay, Credit/Debit Card, and PayPal. Users will also be able to exchange accumulated points for money to be donated to a charity.

1.1 Functional Requirements

- The user logs into the app or creates new account
- They see a map of the area they have selected along with the color codes.
- (They see a color coded map of their surrounding area or a searchable location, maybe also display number of users actively cleaning individual locations)
- They visit a non-green area (either red or uncolored), clean it up
- After they clean it up, they report it as green
- The user utilizes a camera option and takes a picture of an area reported green to confirm
- User gains experience points (points translate to actual monetary value as an alternative to donating actual money amounts) and levels up
- The system will provide screens for viewing maps, viewing user profiles with experience and viewing donation pages
- System will provide a screen for communication with other users (also a utility to report users that are falsely claiming to have cleaned an area?)

1.2 Non-functional Requirements

- Maps should update with every submission to prevent double cleaning of an area
- Map updates and payment processing need to be reliable and take a few seconds at most
- Efficiency and integrity of payment processes are high priority.
- App should be portable to Android and iOS
- User login info should be saved unless manually logged out, for usability
- Users can login using social media accounts, such as Facebook and Google.
- If there are multiple users in the same area, allocate points accordingly
- 99.99999999% uptime for a reliable experience, maximal efficiency
- Server processing and data storage will be done in the cloud using AWS

2. REQUIREMENTS SPECIFICATIONS

2.1 Functional Specifications

- User provides user id and it is checked against database, if not found prompt user to create account, if found prompt user for password
- Get location data from users cell phone and send data to the server
- Server generates color coded map with data retrieved from database holding status of individual mapped locations
- Server sends updated map to users cellphone
- App runs in background, alerts user if other users join cleanup effort
- Once done cleaning, user indicates on app, systems receives notification and updates database, location now changed to green and users data updated with points

2.2 Non-functional Specifications

- Location data from last location of user should be stored in case of a GPS malfunction
- Billing information should be stored in encrypted database

3. USE CASES

3.1 Use Case 1

User can use map to see areas near them and (on mobile only) mark them as complete before/during/after they have picked up litter in their area. User can view their points balance and badges once they log in.

ACTOR

- A user that is using the mobile app.

PRECONDITIONS

- User is registered
- User logs into the app or website to show map view, badges and points balance
- User must have location services enabled

POSTCONDITIONS

- Users' progress data of cleaned data is recorded in the system
- Map is updated to show cleaned area, task to be confirmed by another user (see Use Case 3)
- Profile has a pending reward of points and badges/medals that are dependent upon another user confirming that the area was cleaned

FLOW OF EVENTS

1. User logs in while outdoors
2. User marks area they have cleaned by drawing the area that they cleaned.

3. Map changes to show that the area was cleaned and is pending confirmation.
4. User communicates with other users to challenge them to clean
5. Another user will have to confirm that the area was in fact cleaned (See Use Case 3)
6. Upon confirmation, system updates events and profile. The user that cleaned earns a badge and points are added to their profile.

PROTOTYPE

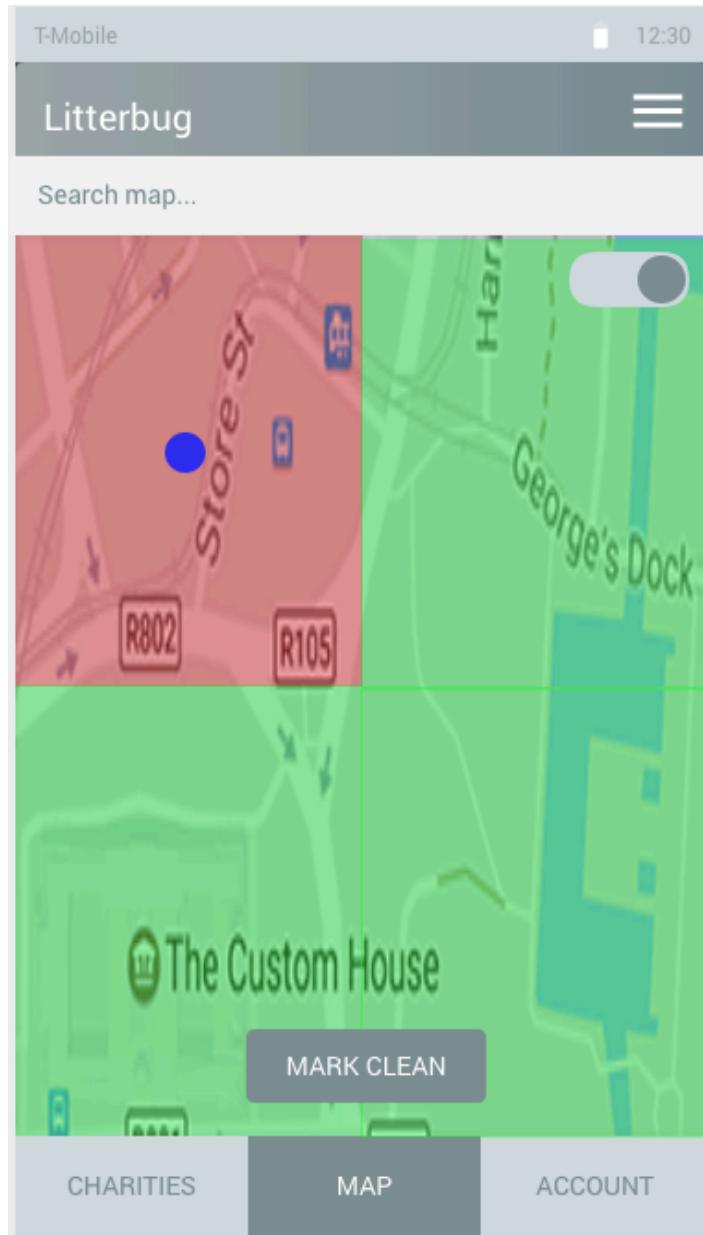


Fig. 1. User views map that shows areas that are clean and areas that are dirty.

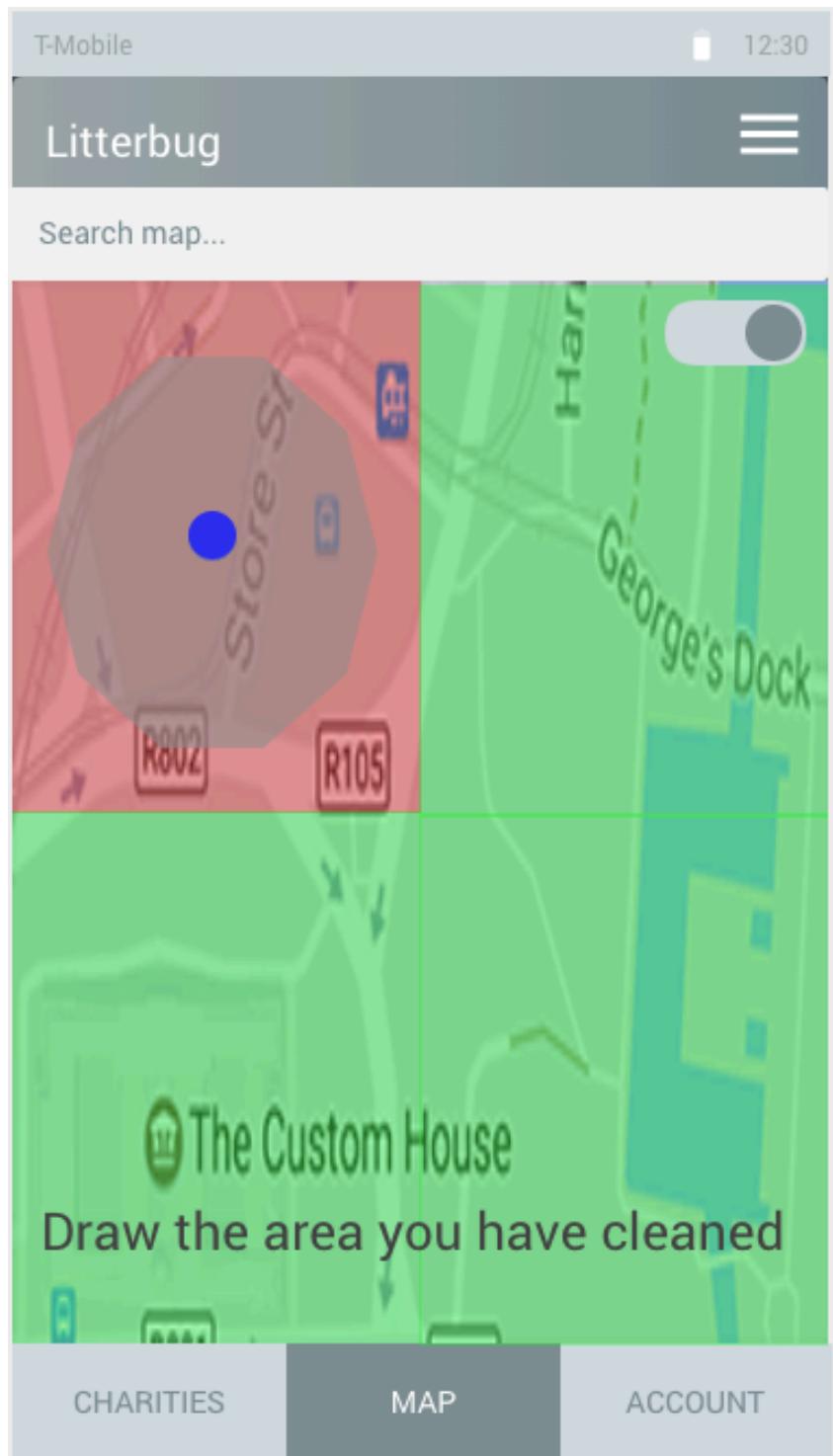


Fig. 2. User taps “Mark Clean” button and is prompted to draw the area they cleaned.

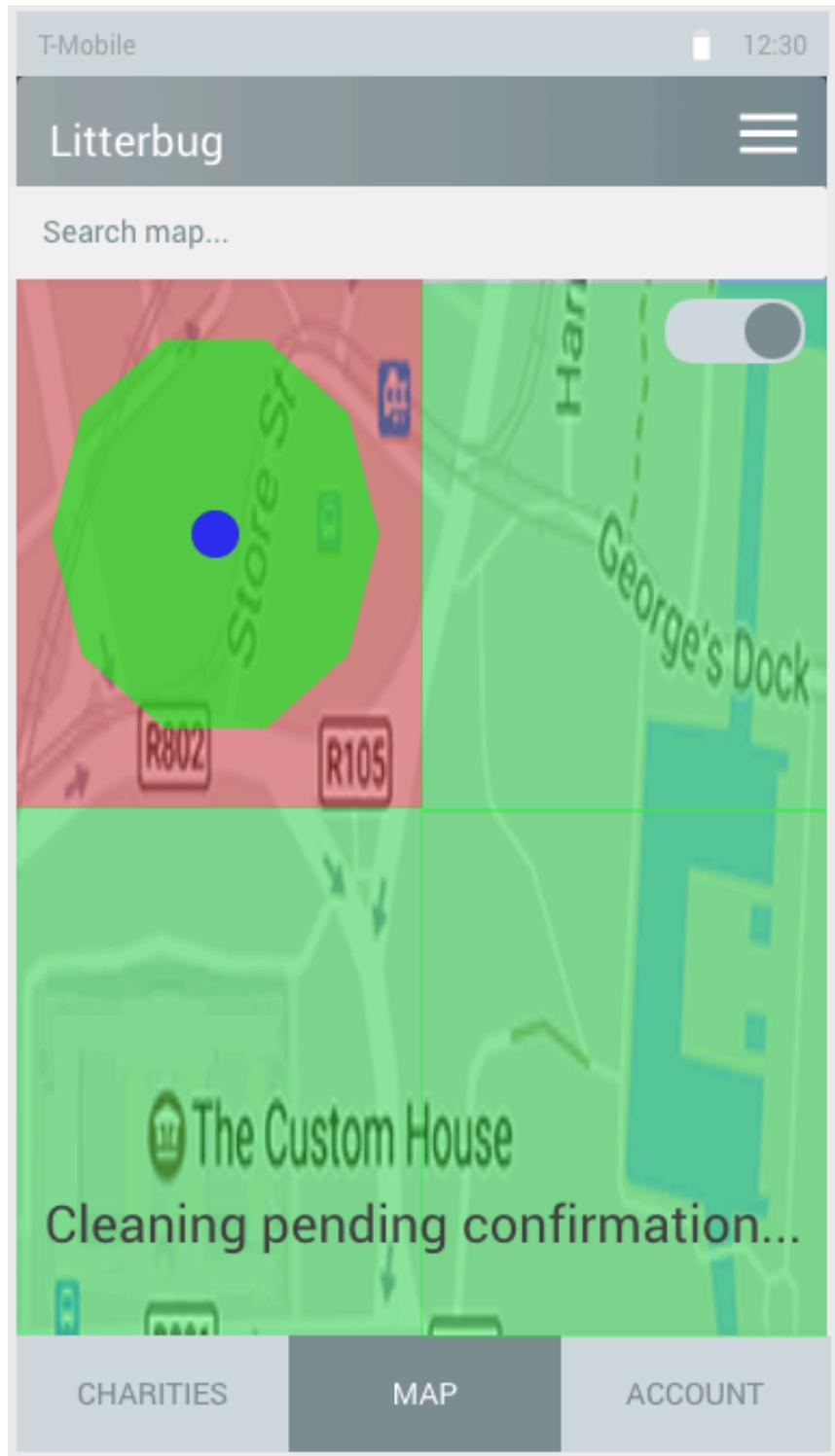


Fig. 3. User draws cleaned area and submits, then waits for confirmation from another user.

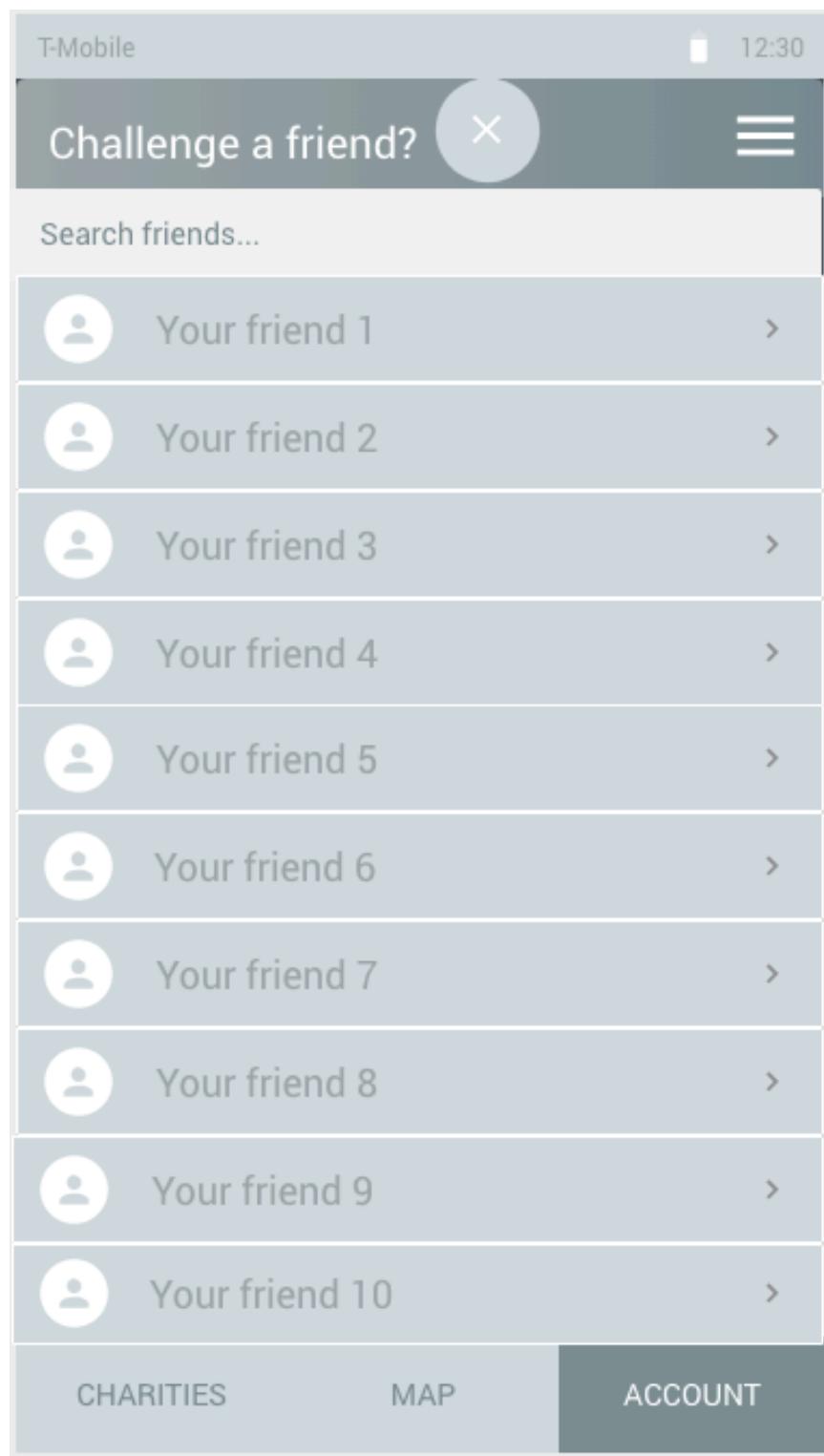


Fig. 4. User selects a friend or another user to challenge them to clean an area.

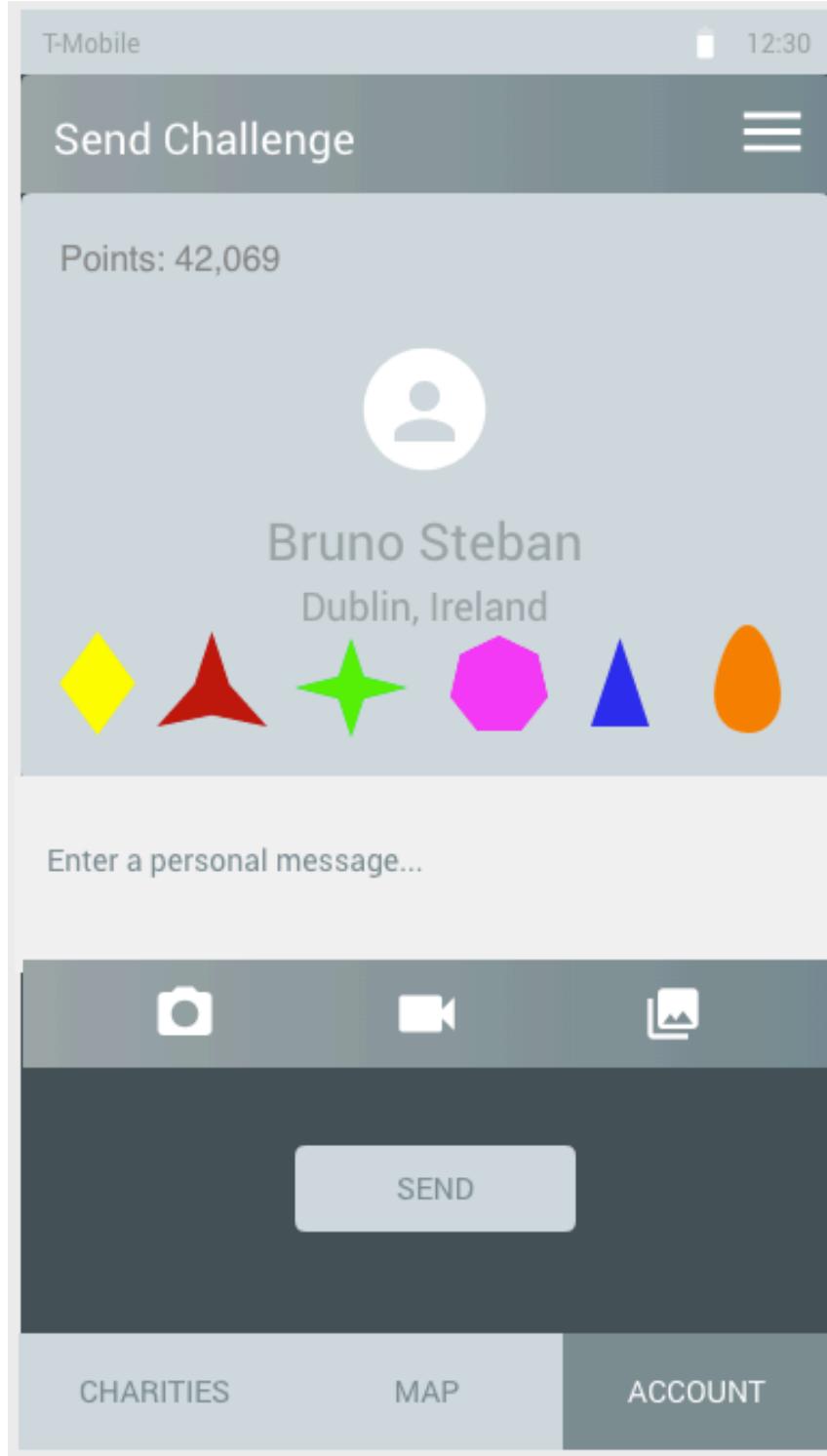


Fig. 5. User can personalize the challenge and include a message or attach a photo/video to show them the area they cleaned or an area that needs cleaning.

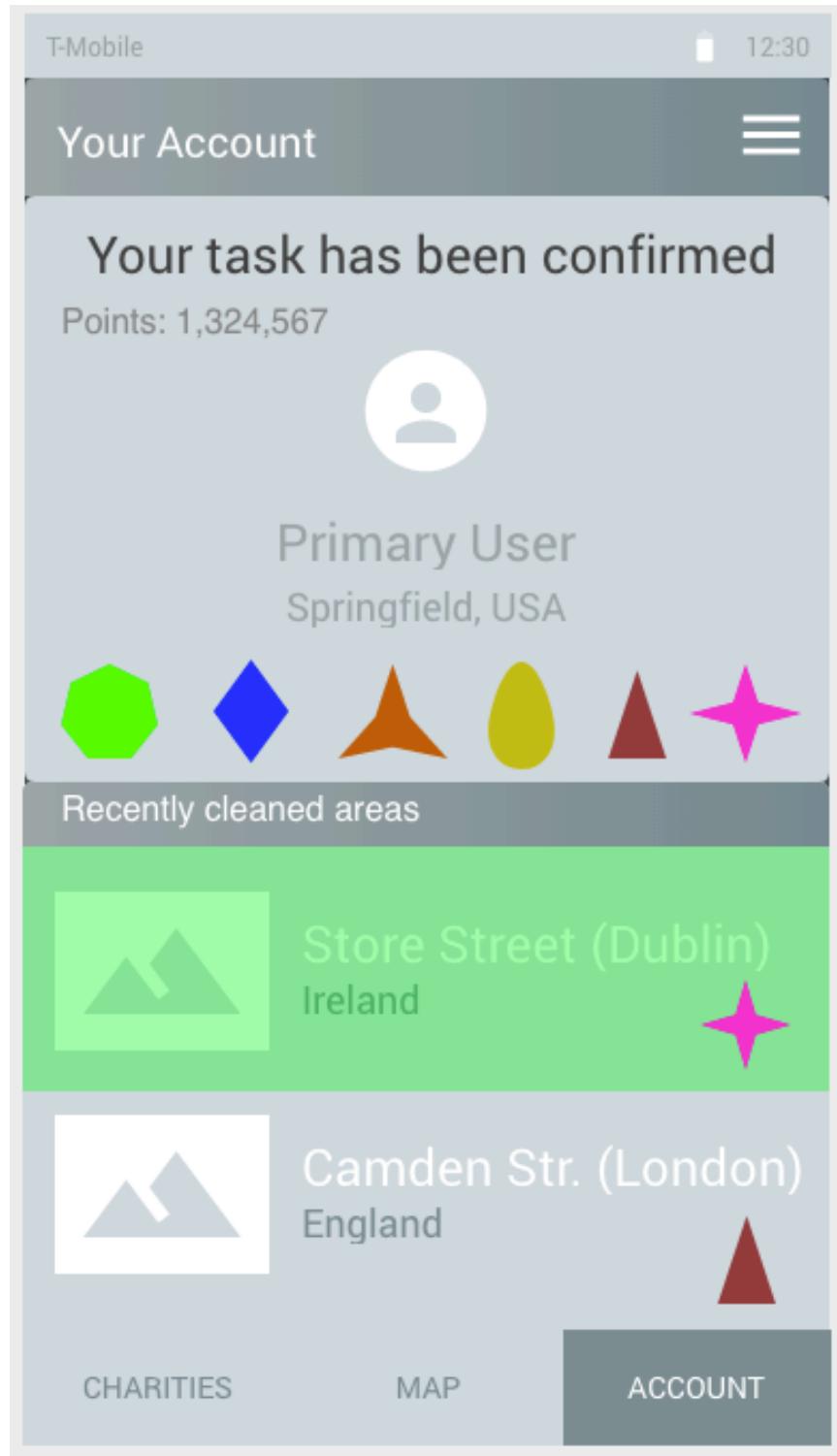


Fig. 6. After another user confirms that the area was cleaned, original user's profile is updated to include newly-earned badges and the recently cleaned area.

3.2 Use Case 2

User can donate to charities, not only at sign up, but at any point during the life of the app. They will get bonus points and medals for doing so.

ACTOR

- A user that is using the mobile app.

PRECONDITIONS

- User is registered and logs into app
- User has Apple/Google Pay setup or has entered credit/debit card information in app
- User has PayPal setup
- User can view accumulated points to donate

POSTCONDITIONS

- User has made a donation
- User exchanges their points for money (at a rate to be determined) to be donated to the charity of their choice.
- Points are deducted from the account after being used for the donation
- User receives a badge/medal based on the amount of donation

FLOW OF EVENTS

1. User logs into app
2. User chooses a charity
3. User chooses a payment method and amount
4. User donates to that charity
5. Charity confirms receipt of donation
6. System updates data
7. User receives reward in form of bonus points and badges/medal(s)

PROTOTYPE



Fig. 7. User Logs into app and chooses a charity.

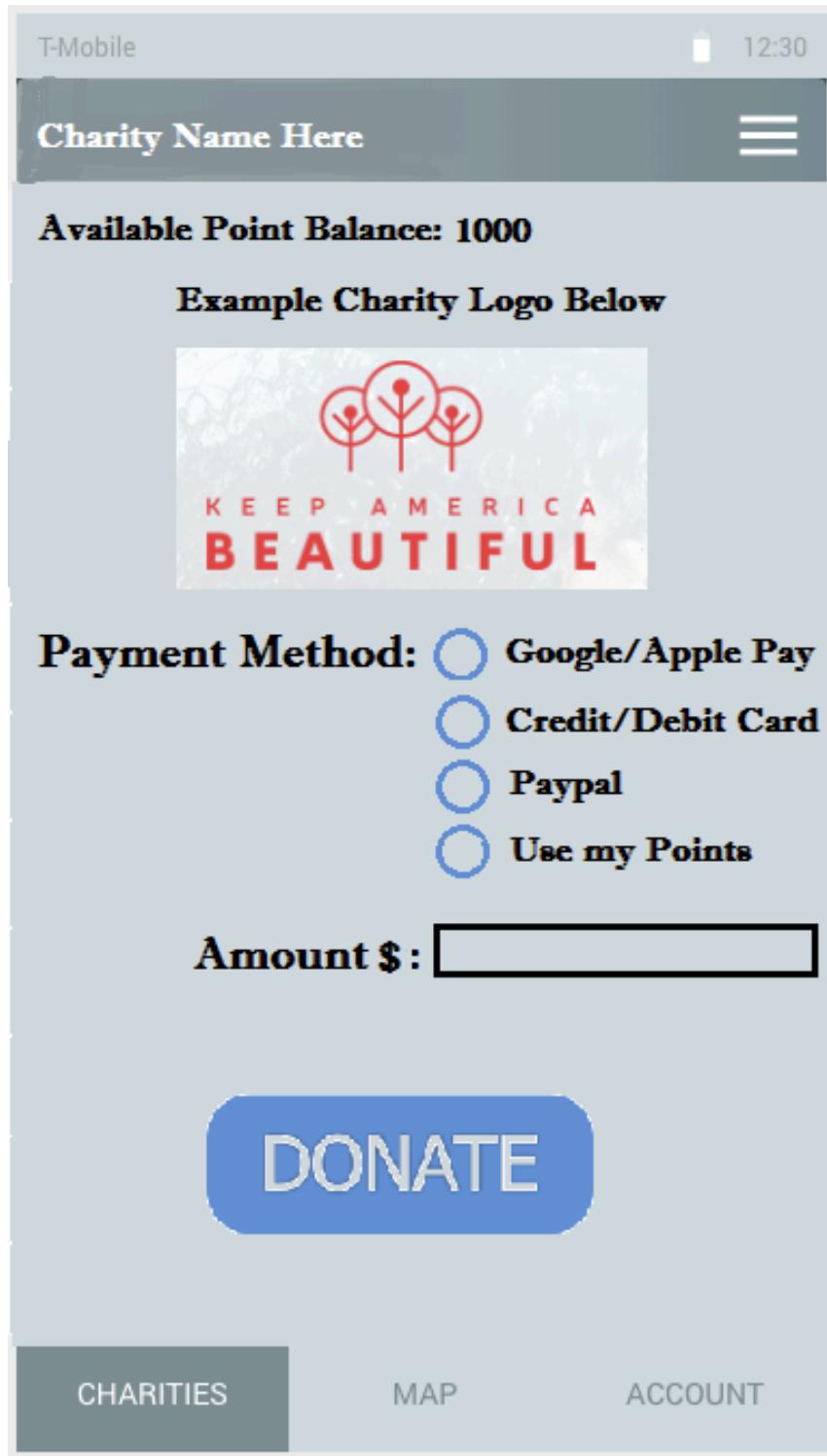


Fig. 8. User chooses a payment method and amount.

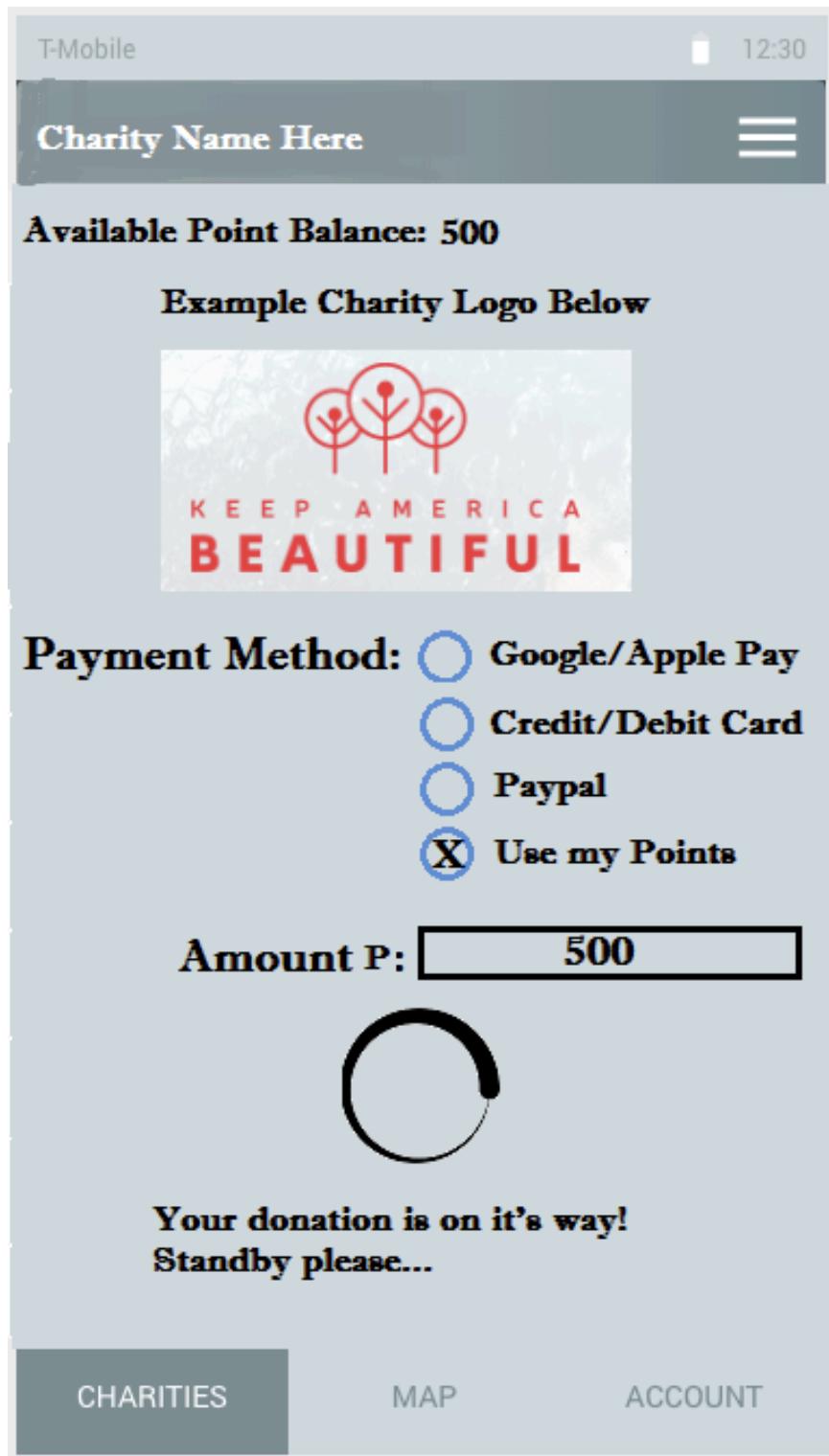


Fig. 9. Donation sends and user waits for confirmation.

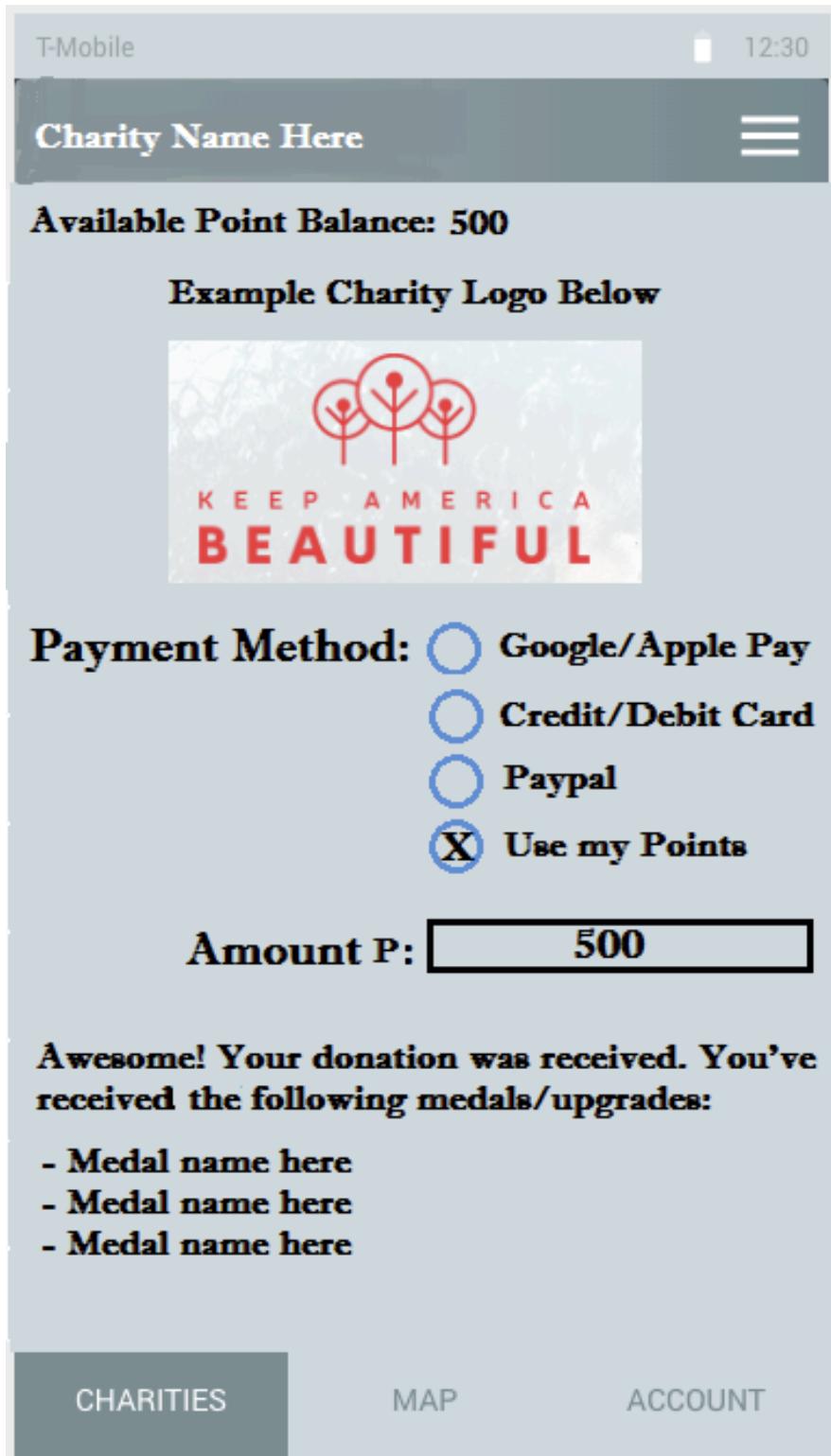


Fig. 10. Charity confirms donation received and user receives reward in form of bonus point and medals.

3.3 Use Case 3

User verifies that cleaned areas are actually cleaned or not and the app leaves the area green or updates it to red if not clean.

ACTOR

- A user that is using the mobile app.

PRECONDITIONS

- User is registered
- User is logged into app and has location services enabled
- User is in green area
- To prevent abuse, check if the user consistently verifies areas that are always/frequently cleaned by another user via photo

POSTCONDITIONS

- Green area changed to red or uncolored area if the area is not actually cleaned
- Last user to mark area as green is flagged for abuse
- User profiles are updated to reflect the cleaned area and feedback is saved to their profile

FLOW OF EVENTS

1. User is logged into app
2. User is currently in an area that is coded green by database
3. User reports area as clean or not clean
4. Database updates map
5. Database finds last user to have cleaned area
6. Database compares time elapsed between cleaning and reported condition to determine if reasonable time has passed for area to become dirty
7. If timing is unreasonable database flags previous user, when enough flags are raised delinquent user is issued warning or removed from system

PROTOTYPE

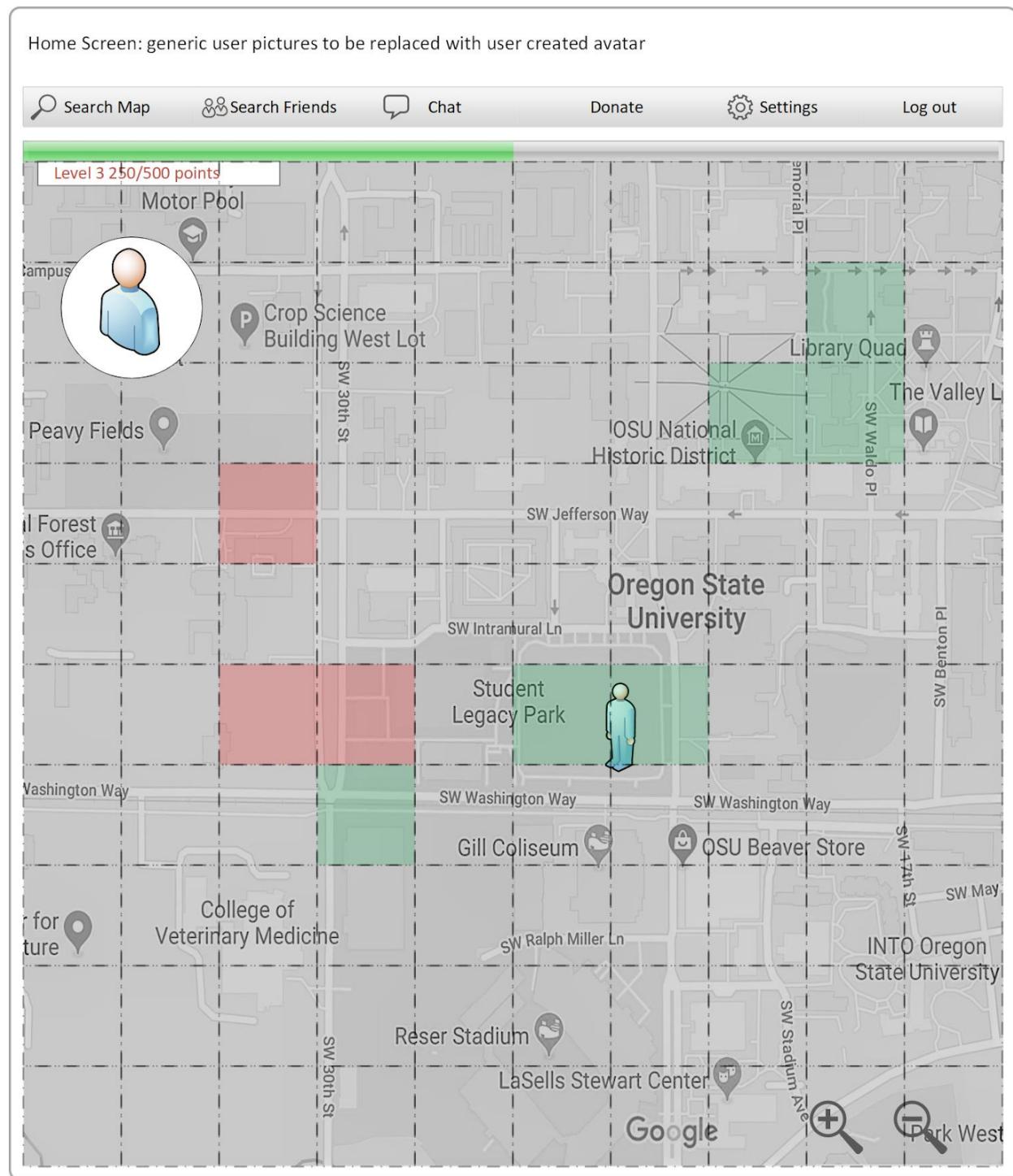


Fig. 11. User opens app and views map.

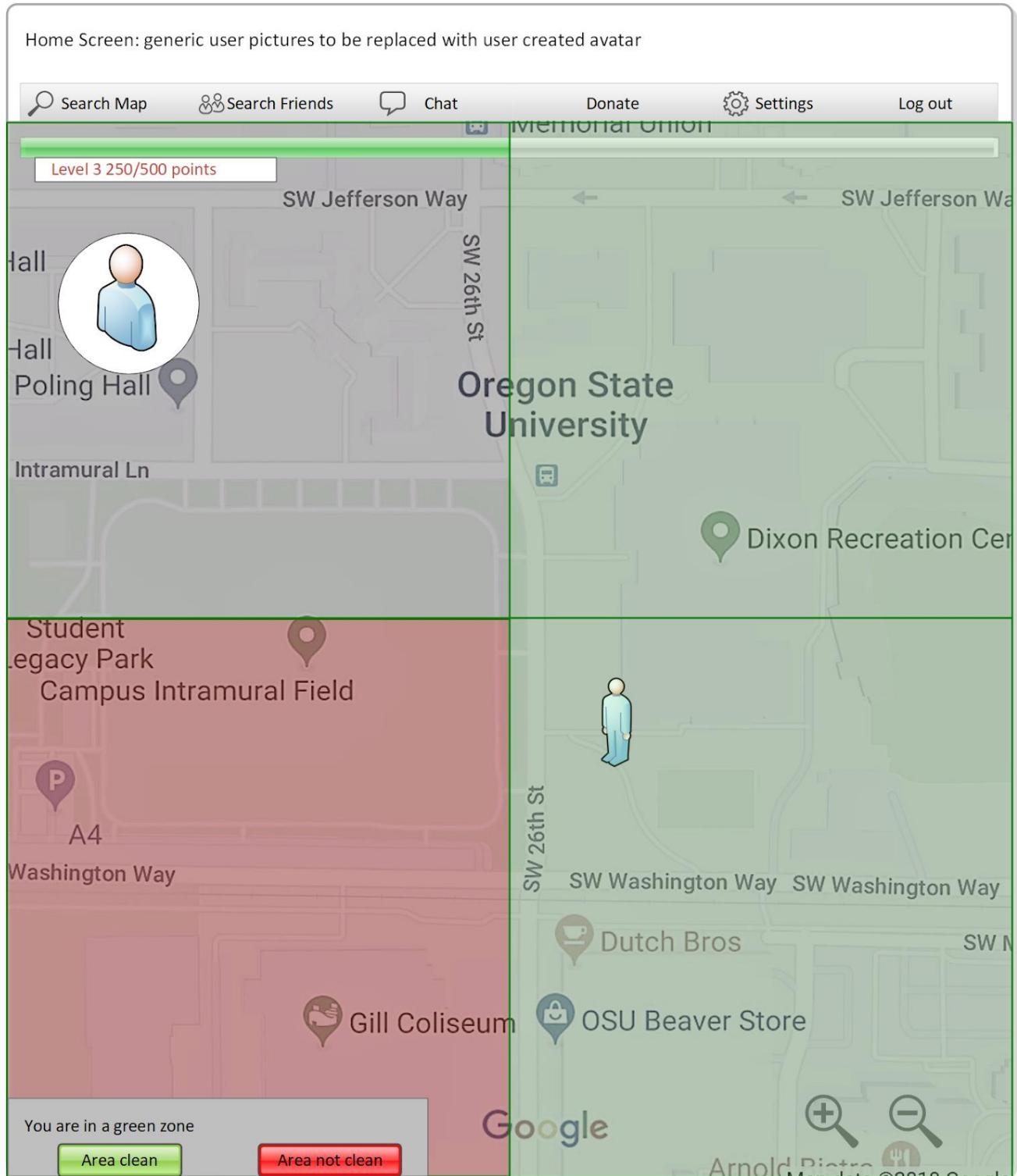


Fig. 12. User is in a green area to confirm that area has been cleaned.

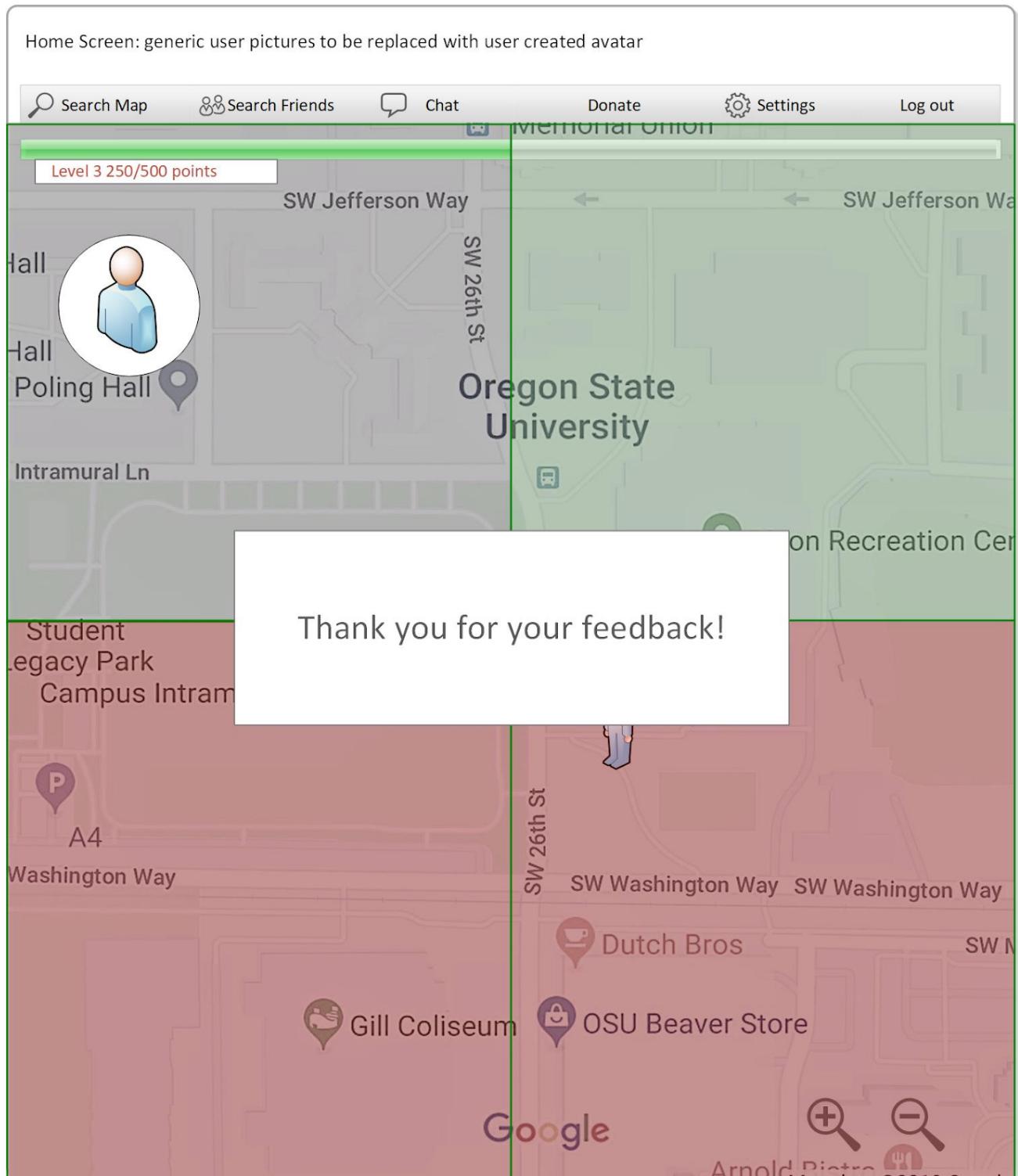


Fig. 13. User marks the area as cleaned or reports it not cleaned.



Fig. 14. If a user submitted an area that they cleaned but another user marked it as still dirty, the original user gets a warning.

4. ENTITY RELATIONSHIP DIAGRAM

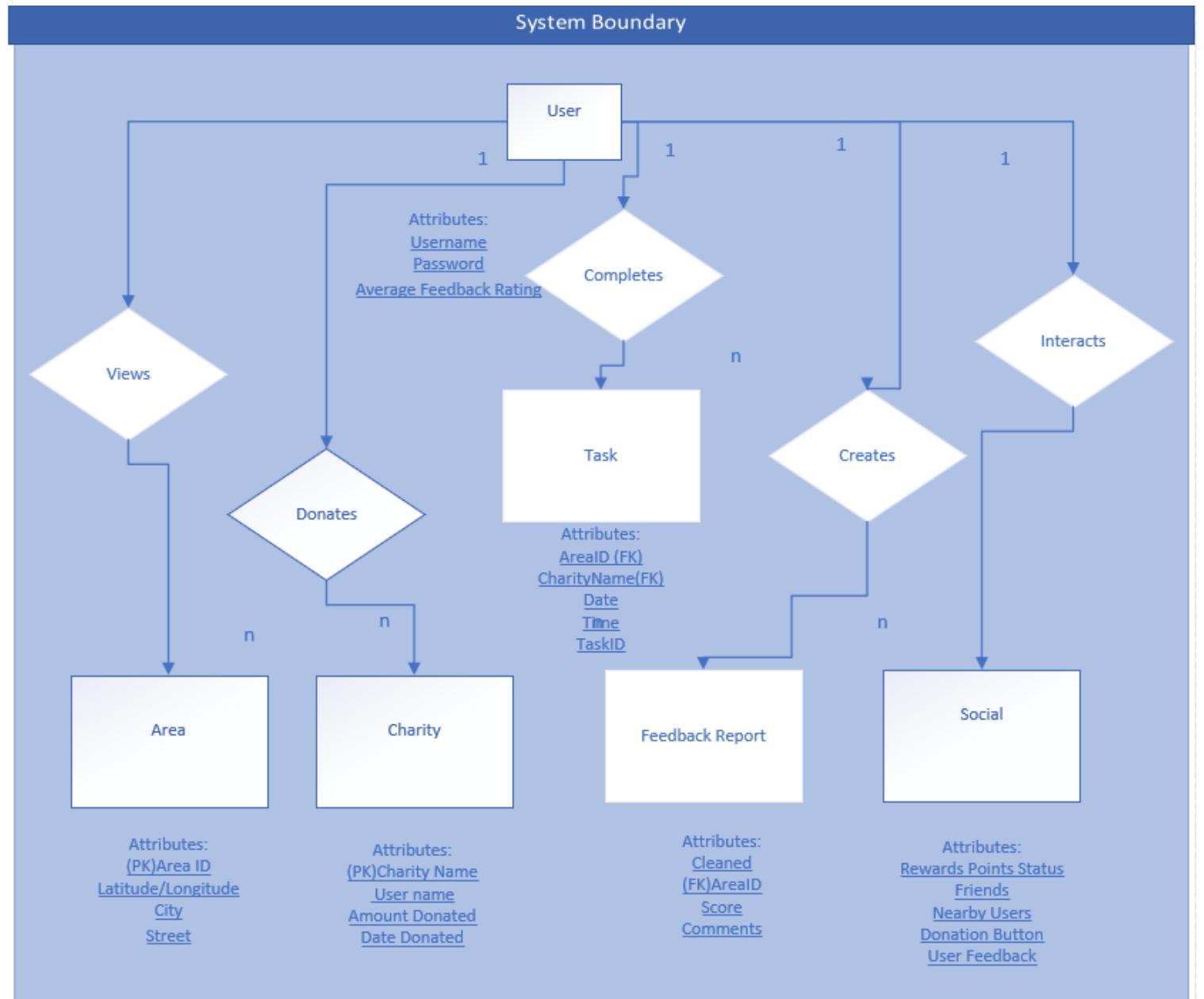


Fig. 15. Entity Relationship Diagram with the following entities:

- User
- Task
- Area
- Charity
- Feedback Report
- Social Profile

5. DATA FLOW DIAGRAM

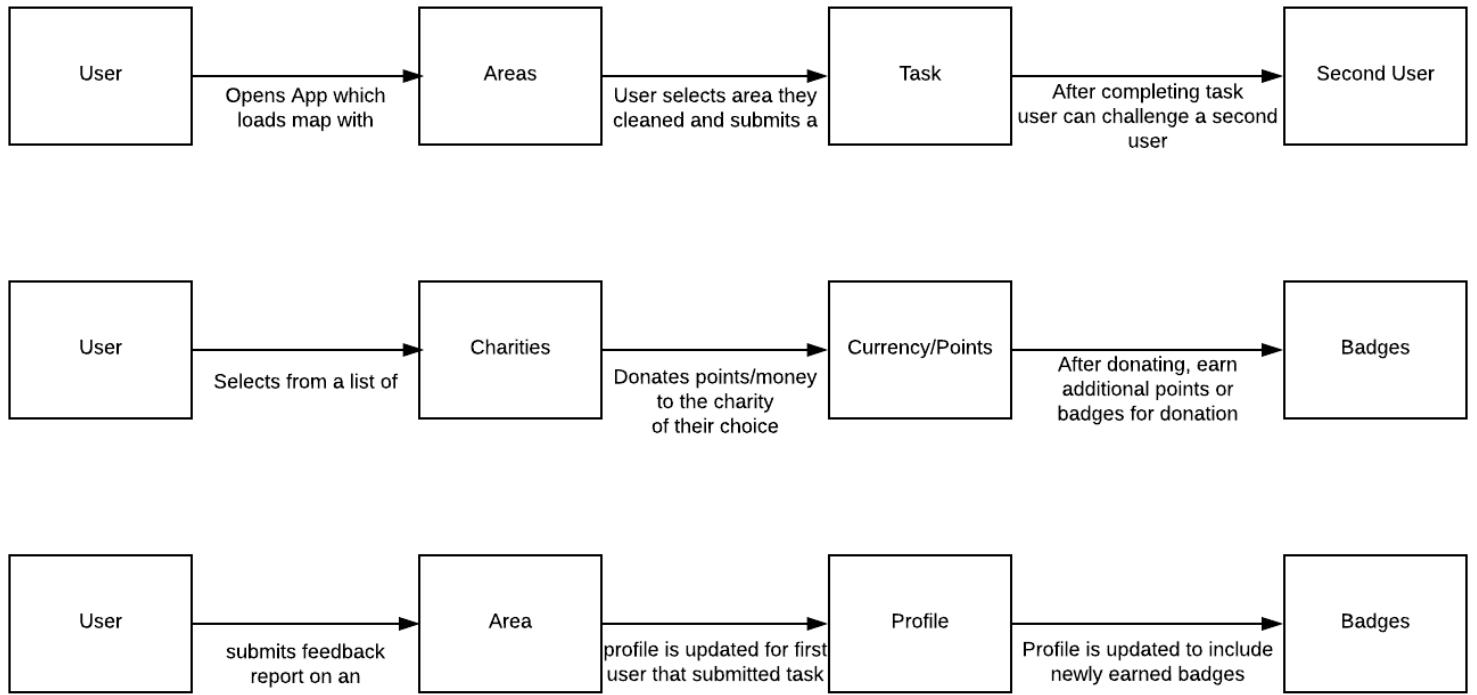


Fig. 16. Data Flow Diagram that shows the flow of data for each of the 3 use cases.

6. MESSAGE SEQUENCE DIAGRAMS

6.1 Use Case 1

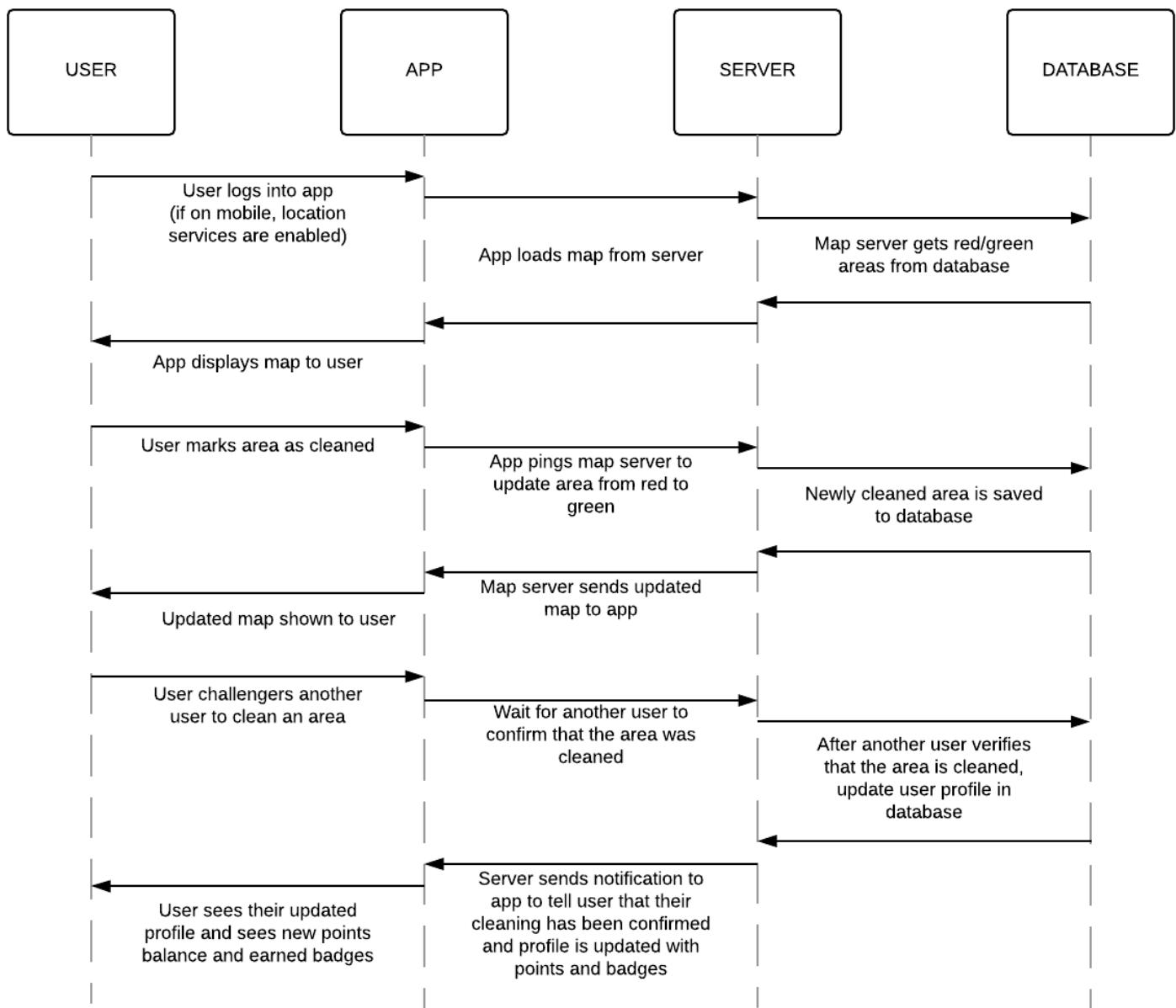


Fig. 17. Message Sequence Diagram for Use Case 1

6.2 Use Case 2

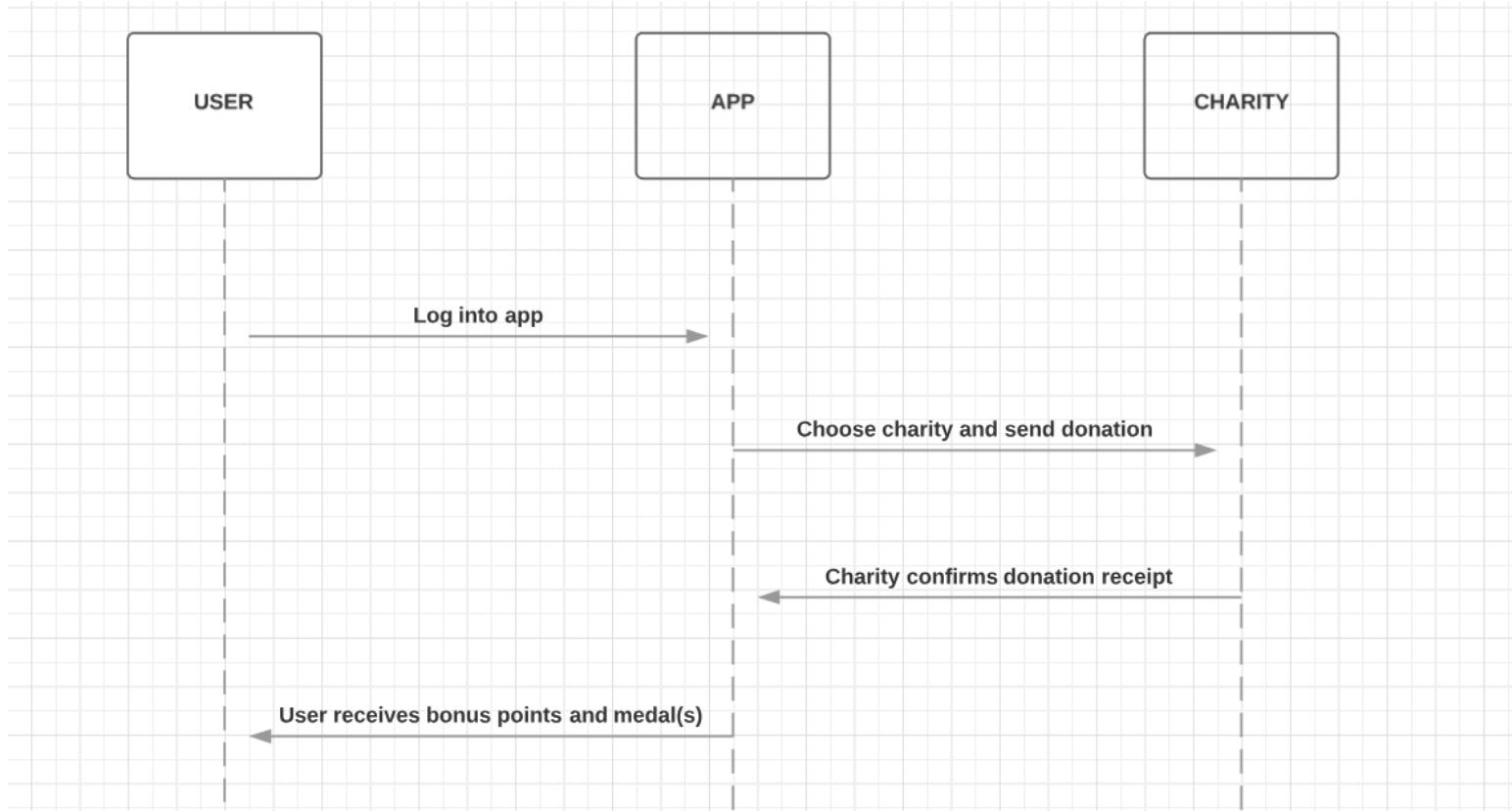


Fig. 18. Message Sequence Diagram for Use Case 2

6.3 Use Case 3

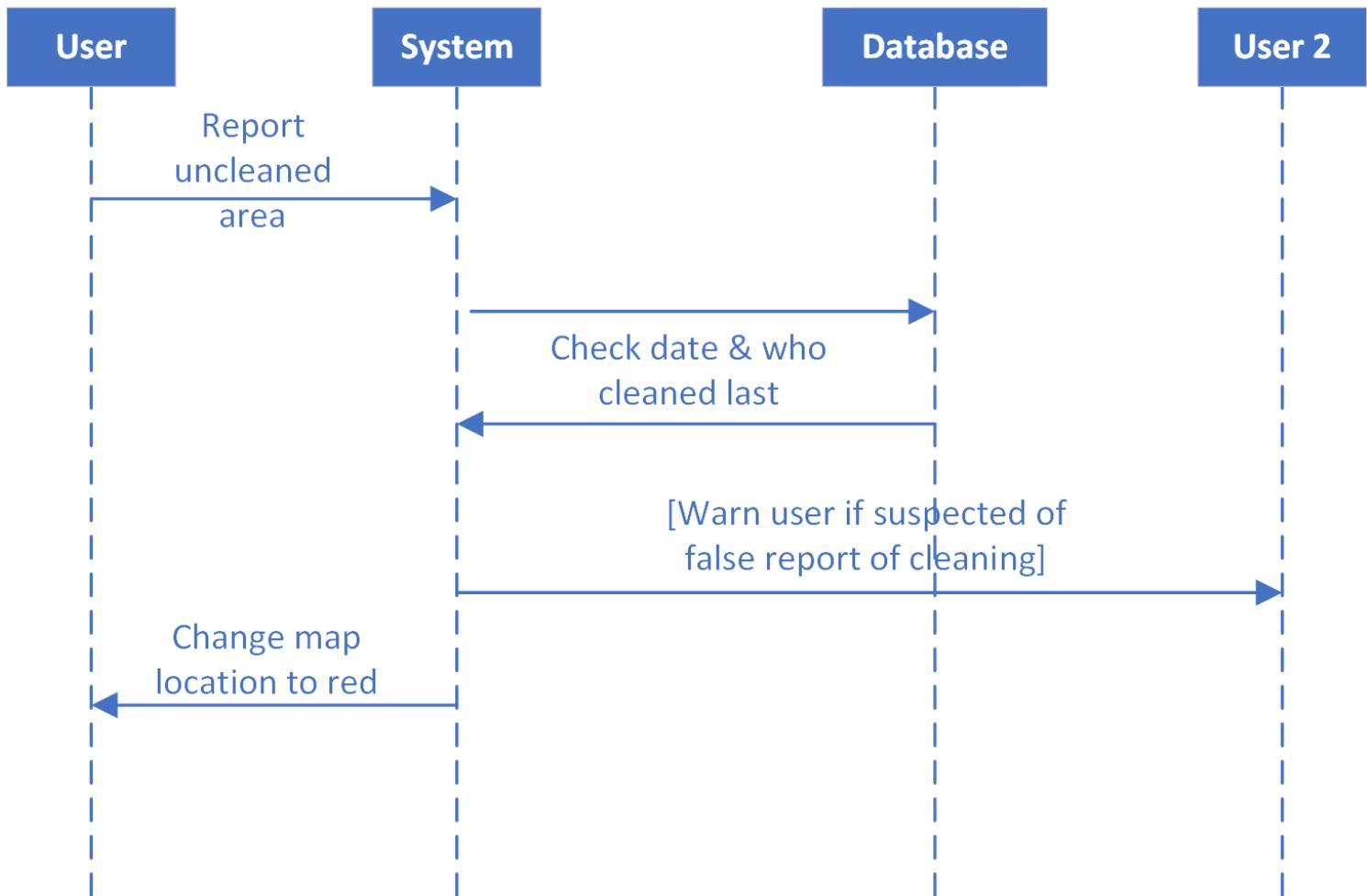


Fig. 19. Message Sequence Diagram for Use Case 3

7. CHANGES SUMMARY

7.1 Requirements Definitions

FUNCTIONAL REQUIREMENTS

1. “User can view their points balance and badges once they log in.”
 - One of the main purposes of this application other than cleaning litter, is the ability to donate. We felt that the display of the badge upon login would compel the user to progress for more badges by cleaning litter. In addition we felt that the display of points to the user would compel them to donate more frequently since they have the point balance easily accessible. We felt these two UI improvements furthered the purpose of the application.

2. "User gains experience points (points translate to actual monetary value as an alternative to donating only money amounts) and levels up."
 - We needed a method where a user would gain points while a charity would also benefit. It could potentially be quite expensive for the user if they simply donated using conventional currency. To remedy this problem we proposed that points would equate small currency increments e.g. 1 point equals 10 cents. This way a user can gain credit in the system, without the practice of participating in the application becoming prohibitively expensive.
 - Previously, we proposed we would allocate points for the amount donated. Meaning a user would receive points for money donated. Instead we are decided to reward points to the user for tasks completed, then to have those points translate to a monetary amount once the user was ready to donate. After points have been converted to donations to charity, the user receives an upgrade to their merit badge.
3. "The user utilizes a camera option and takes a picture of an area reported green to confirm."
 - This was added as a graphical feature to confirm that cleaning was done by other users. This is a necessary feature to hold abusers accountable for their work.
4. We elected to add a PayPal method of payment
 - Having several secure methods of donation available on the app made us believe that the more options of funds the user can see, the easier it is for the user to donate, and the easier it is to donate to charity, the more likely the user will do it.

NON-FUNCTIONAL REQUIREMENTS

- No changes were deemed necessary.

7.2 Requirements Specifications

FUNCTIONAL SPECIFICATIONS

1. "If user zooms in, the squares disappear and the exact areas cleaned will be outlined"
 - We realized we needed a more detailed method of seeing which areas were cleaned since users do not necessarily clean in perfect squares when using this app e.g a walk on the beach or trail. To remedy this, we allowed for the zoom feature to remove square displays and show exact paths drawn by users of the areas they have cleaned.

NON-FUNCTIONAL SPECIFICATIONS

- No changes were deemed necessary.

8. SUMMARY OF WORKING WITH THE CUSTOMER

Michael Johnson was able to meet with us on Thursday, July 12th. He was mostly happy with our manifestation of his vision and requested changes that were quite minor. He was able to answer any questions or clarifications we had and helped us get closer to his original vision. The changes we made are outlined above and cover the Requirements Definition and changes to all 3 Use Cases.

9. SUMMARY OF GROUP MEMBER CONTRIBUTIONS

- David Mednikov
 - Updated requirements, prototypes, and message sequence diagram for Use Case 1, converted document from Google Docs to ACM Format
- Hitesh Varma
 - Updated requirements definitions and requirements specifications per feedback from the customer, created Google Doc for collaboration on HW2, described changes made to requirements definitions and requirements specifications from HW1 to HW2
- Alexander Yfraimov
 - Updated Use Case 2 requirements and prototypes to include support for PayPal as a payment method and to display user points balance on various screens
- Kevin Allen
 - The customer was happy with the prototypes and requirements for use case 3, so changes were not necessary