Ripple
Requirements and Specification Document
2020-02-15, version 1.0

Project Abstract

Ripple will be a reporting app to spread and bring awareness to discrimination and violations of rights (e.g., racial equality, rape, abuse, gender, Black Lives Matter). Through live data accumulated reported by people here in the U.S. and the world, violations can be visualized, mapped, and reported to instigate change. Users can create reports which are then posted as violation incidences on a map. Areas of hotspots/heatmaps, time/dates, locations will be visualized on the map. This project will be exciting to implement in that it will use many technologies, full stack, mobile, database, cloud, data visualization, APIs, and best of all you will be developing an app to fight the scourge of racism. Give a voice to the oppressed!

Document Revision History

Rev. 1.0 <2021-02-16>:initial version

Customer (Penghai)

Our application will be delivered to every one who feels passionate fighting against discrimination. Since the application both includes functions for people to report such incidents and to view them, people who wants to avoid the areas or places reported with discrimination events will also benefit from our application. Criminal incident on the basis of discrimination will be be reported to the police department in some way (pending privacy rules) so the police is also a potential customer. The specific client that will bring specific requirements and negotiate with us will be Professors LiLi Johnson and Jess Waggoner of UW-Madison Gender Women's Studies. They can provide some insights into inequalities in disability and digital contexts of women, gender, race equality studies which are helpful to our development process and testing procedure.

Competitive Landscape (Penghai)

The possible competitor in the market will have some of the following features: considerably large user pool such as Facebook or Tiktok, strong development team that can copy our idea before we occupy the market, and similar functionality with our core feature. Competitors can take advantage of a large user pool and strong development team to copy our idea before we can occupy the market with stable users since it is not strongly based on copyrights and patents. However, it is possible for us to specialize our application with knowledge from expertise in the area of ethic study. In this way, we can obtain the unique advantage of carefully crafted, professionally advised features such as suggestions for users to deal with discrimination or to judge if something is biased that we can add to our application to prevent copying. Also, there are applications out there that have some similar functionality. For instance, the alert message that we receive on our phone via mobile service provider and the local police department is one similar product. However, it only shows the most severe cases that might include criminal act

while our application delivers all related cases with discrimination that are easily viewable via the google map API. Therefore, our main advantages against competitors with similar products will be convenience of using and varieties of functions which we will value in the development process.

User Requirements (Kerui)

1. Report discrimination incident
    a. Select incident type
        i. Can search by disability, gender, and other forms of discrimination
    b. Select incident details
        i. Gives dictionary definitions of what terms like “harrassment” or “indirect discrimination is”
    c. Select incident Location
        i. Can search by buildings or cities
    d. Confirm report
2. View discrimination incident map
    a. See incident information in local map
        i. Selection a location pin for detailed incident information
    b. Search incident
        i. By location
        ii. Filter by time
        iii. Filter by discrimination type
        iv. Filter by concentration (heat map)
    c. View time trends as a whole on map
    d. View heat map as a whole on map

Possible Additional User Requirement: View Discrimination Information Screen

a. Users, without needing to report, can view specific definitions for different types of discrimination (somewhat educational)

Use Cases (Sunjun, Titus)

Use cases that support the user requirements in the previous section. Every major scenario should be represented by a use case, and every use case should say something not already illustrated by the other use cases. Diagrams (such as sequence charts) are encouraged. Ask the customer what are the most important use cases to implement by the deadline. You can have a total ordering, or mark use cases with “must have,” “useful,” or “optional.” For each use case you may list one or more concrete acceptance tests (concrete scenarios that the customer will try to see if the use case is implemented).

Use Cases: Report a discrimination incident, view discrimination incidents for a location, view discrimination map for a region, view discrimination trends (category (only racial discrimination), time trends, heatmap/density)

Name	Report a discrimation incident
Actors	Users who are suffered discrimination
Triggers	He/She is discriminated against
Events	When users encounter violations of rights (e.g., racial equality, rape, abuse, gender, Black Lives Matter), they upload the incident type, detail, location through the App by 'Report Discrimination'and select a 'Location' then 'Confirm Report'.
Exit condition	·The violation case is successfully uploaded · After Submitting and entry will take the user to the corresponding location on the map. ·The incident can be searched through the APP
Post conditions	The specific violation report (type/location/detail) is visualized, mapped, and reported through the APP.
Acceptance test	Check that the discrimination case has been in the APP library: 1.User clicks to view discrimination map 2.User choose the incident on the location 3.The incident details are displayed correctly 4. The record is stored in the back-end database and the database updates successfully.

Name	View discrimination incidents for a location
Actors	APP Users
Triggers	Users want to know about discrimination incidents in a particular location
Events	Users use the APP to view specific details,

	type,location of discrimination incidents in a location by clicking ‘View Discrimination Incident Map’and search incidents by location.
Exit condition	Users can see the discrimination incidents in a certain location and they could see the type and details of these incidents by clicking.
Post conditions	·Users could exit the map successfully by clicking ‘Leave Map’. ·The act of viewing does not change the map or the data ·Viewing the case again, get the same incident type/details. · As discrimination cases rise or fall, the list of cases in a location is updated.
Acceptance test	Check that the discrimination incidents for a location could be viewed: 1.User clicks to view discrimination map 2.User search locations 3.The incident list is displayed. 4.Click on the specific incident, the type, details could be viewed. 5. The record is stored in the back-end database and the database updates successfully.

Name	View discrimination map for a region
Actors	APP Users
Triggers	Users want to view the discrimination map for a specific region
Events	Users use the APP to view the map of the discrimination distribution on different locations in a specific region by clicking ‘View Discrimination Incident Map’.
Exit condition	Users see an overview map of discrimination at different locations in a certain location.
Post conditions	·Users could exit the map successfully by

	clicking ‘Leave Map’. ·The act of viewing does not change the map or the data ·The map will be updated as discrimination cases are updated.
Acceptance test	Check that the discrimination map for a region could be viewed: 1.Users enter the APP 2.Users click the ‘View Discrimination Incident Map’ button 3.The region map is displayed with some general information of the decrimation incidents. 4.The map updates as the records on the database updates. 5. Exit by clicking ‘Leave Map’ successfully.

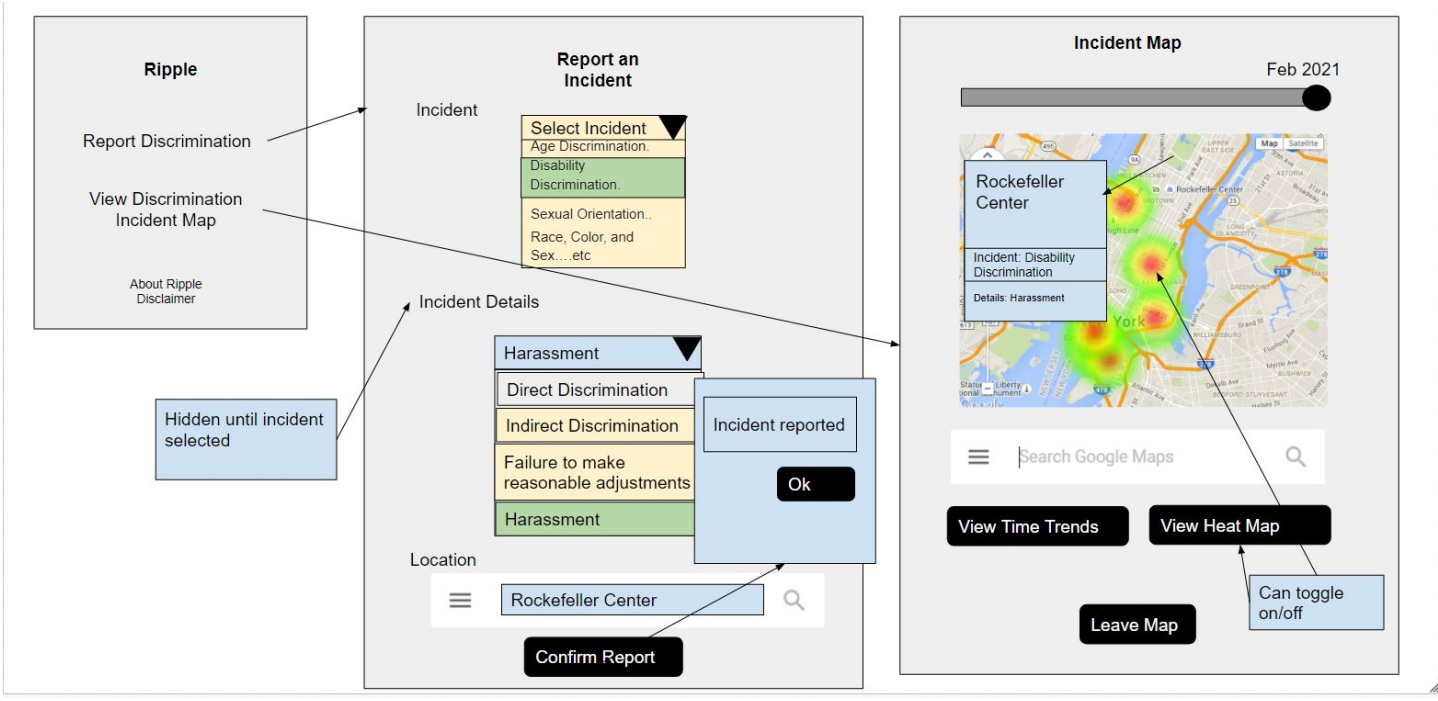
Name	View discrimination trends (category (only racial discrimination), time trends, heatmap/density)
Actors	APP Users
Triggers	Users want to view the discrimination trends
Events	Users use the APP to view the discrimination trends(category (only racial discrimination), time trends, heatmap/density) on different locations in a specific region by click ‘View Discrimination Incident Map’ button then choose the ‘View Time Trends’ or ‘View Heat Map’.
Exit condition	The user successfully see the time trend or heat map of the discrimination incidents at some locations in a certain region by searching and filter.
Post conditions	·Users could exit the map successfully by clicking ‘Leave Map’. ·The act of viewing does not change the map or the data ·The time trends will be updated as

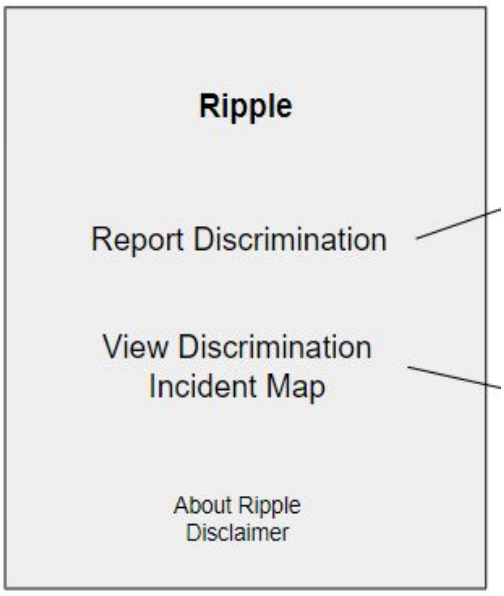
	discrimination cases are updated.
Acceptance test	1.Users enter the APP 2.Users click the 'View Discrimination Incident Map' button 3.Users click the 'View Time Trends' or 'View Heat Map' button 4.The time trends are displayed with some general information of the decrimation incidents.

User Interface Requirements (Titus, Eric)

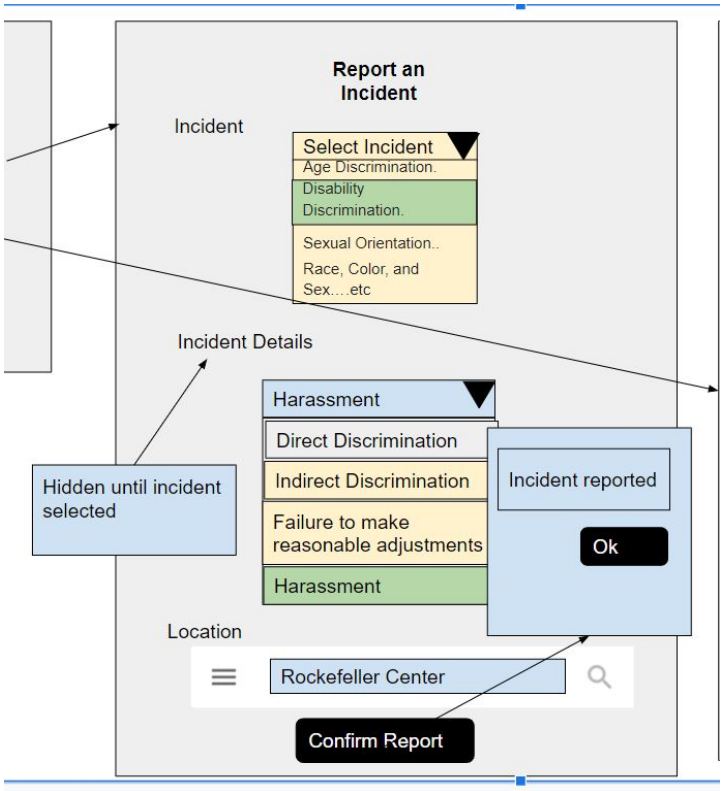
Describes any customer user interface requirements including graphical user interface requirements as well as data exchange format requirements. This also should include necessary reporting and other forms of human readable input and output. This should focus on how the feature or product and user interact to create the desired workflow. Describing your intended interface as "easy" or "intuitive" will get you nowhere unless it is accompanied by details.

The goal of this user interface is to make it easy for users to report and view incidents of various forms of discrimination. To meet this goal, we will have both reporting and map view tabs.

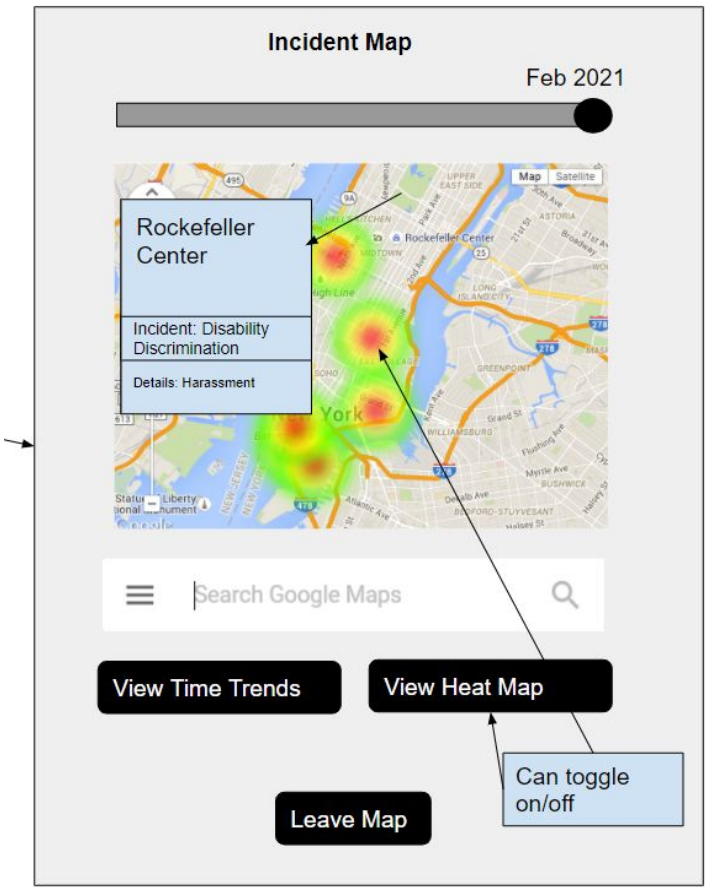




From the home screen the user can navigate to create a report of discrimination or view the world map visualization of all reported discriminations. They can also see information about the app and the disclaimer. Clicking on “Report Discrimination” leads to the image below “Report and Incident” pane. Clicking “View Discrimination Incident Map” leads to the “Incident Map” pane (image below the report incident pane). Clicking “About Ripple” will give info about the app. Clicking “disclaimer” will give info about careful use, honor code of using the app.



Selecting the “report discrimination” option will lead to the report an incident pane, from here the user can select from a dropdown the incident. This incident selection will make visible the incident details dropdown which will have option details related to the incident selected. The user can enter in and then select a location for the incident. After all of this, the user can confirm the report (“Confirm Report”) and will be prompted with a completed report of incidence popup.



Once the user confirms the report they are taken to the map view to the location of the report. They can view time trends (this will show trends with respect to the types of reports, details), adjust time span with the draggable tool on the top (this will affect the viewable data points of interest on the map), search/view locations of other incidences in the search box, and toggle on/off heat maps of all incidences around the world. Users can also leave the map view and return back to the home screen.

Security Requirements (Junyu)

The first security problem that we need to care about is protecting the user’s personal data. We will encrypt all data collected from users, such as the user's IP address, with a Symmetric-key algorithm. Because we decided not to have a user module, which means users do not need to register and login to use our service, we do not need to encrypt other user’s data.

The second security problem is the denial of malicious and non-stop crime reports. We decided to use a Redis counter or Bloom Filter counter to block the malicious IP and reject their potential dos attack.

The third security problem is how to deny fake criminal reports. At this point, we choose to trust users of their reports.

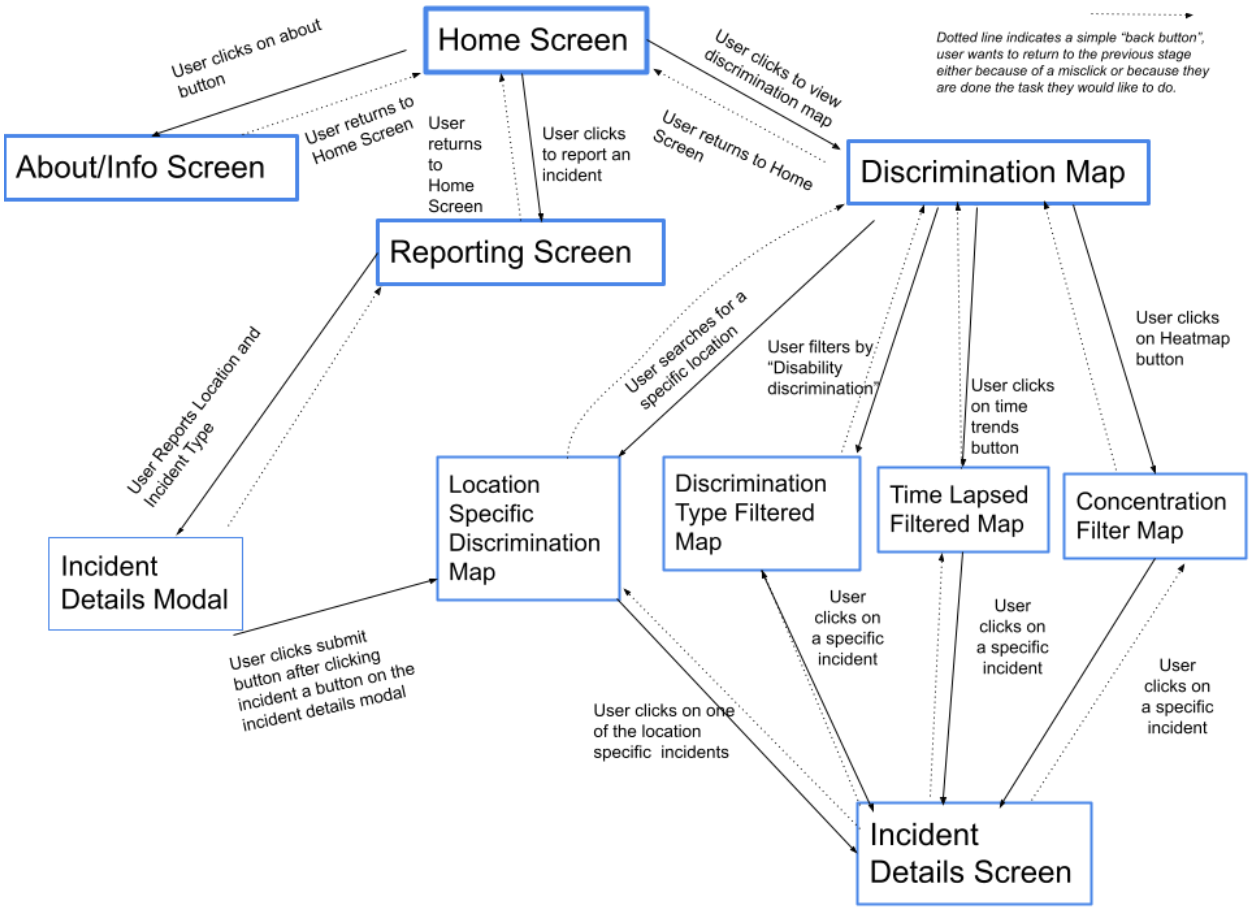
System Requirements(Junyu)
System requirements could be divided into two parts: frontend and backend requirements

For the frontend part (Client), we require users to have an Android/IOS system. The app will use Expo to locally deploy, with the possibility of remote deployment to app stores later on. We also require the users have access to an available network while using the service. With regards to specific app functionality, the app will make heavy use of the Google Map API for visualization of incidents. This will be the major third-party and we will have to adhere to their API format, which will be a restriction. Finally, the App will be written with React Native, which isn't a dependency per se, but a Javascript Framework.

For the backend part (Server), we will run our system on an Ubuntu system on the public internet. This part would only be exposed to the frontend code. The Ubuntu server should have installed python and its related packages (Flask, Mysqldb, Redis-py, etc.). Our system would make sure that the response time of any operation is less than 100ms, and our CPU usage would not exceed 70%. We would also support storing more than 50GB of data, if needed, we could also buy a larger database to serve the increasing user's needs. In the future, if the Query Per Second (QPS) is high, we could add more servers and use Nginx to do the load balancing.

Specification (Titus)
A detailed specification of the system. Every possible execution should be in the specification, though not every aspect need be covered in extraordinary depth. UML, or other diagrams, such as finite automata, or other appropriate specification formalisms, are encouraged over natural language.

Below is a Finite Automata that demonstrates the workflow on the front end. Arrows contain user input and blue boxes are Screens or Modals that the User Interacts with. Dashed lines indicate simple "back buttons" so they do not illustrate the major intended workflow.


Home Screen
About/Info Screen
Reporting Screen
Discrimination Map
Location Specific Discrimination Map
Discrimination Type Filtered Map
Time Lapsed Filtered Map
Concentration Filter Map
Incident Details Modal
Incident Details Screen
User clicks on about button
User returns to Home Screen
User returns to Home Screen
User clicks to report an incident
User clicks to view discrimination map
User returns to Home Screen
Dotted line indicates a simple "back button", user wants to return to the previous stage either because of a misclick or because they are done the task they would like to do.
User Reports Location and Incident Type
User searches for a specific location
User filters by "Disability discrimination"
User clicks on time trends button
User clicks on Heatmap button
User clicks submit button after clicking incident a button on the incident details modal
User clicks on one of the location specific incidents
User clicks on a specific incident
User clicks on a specific incident
User clicks on a specific incident