Disaster relief social network

Ticketing/Post System:

Help → In the context of our domain ***help*** is defined by strict parameters. When a help requester or a help provider request/provide help they are sending or receiving the following:

- Hygiene products (Toilet paper, paper towels, soap, razors, sanitary pads, etc..)
- Water
- Non-Perishable food
- Warm food
- Housing
- Companionship (Asking people for support) Can be in the form of texts or presential meet ups.

Commented [DLMP1]: Help types.

Who are the players?

- Individuals in need of help. (**Help Requesters)**
- Individuals with the capacity to provide help. (**Help Providers)**

The role of each player is **INTERCHANGABLE.** What does this mean?

- A user can at any moment request help or provide help. Their role is assigned to them when they **create a ticket or accept a ticket.**
- In other words, a user will begin in a **neutral state.**

Commented [d2]: Need to edit this.

Help Request Ticket→ A ticket is like a social media post. It will contain the following:

Commented [d3]: Work on this!!! There are two types of ticket now!!! Provider Ticket and requester ticket.

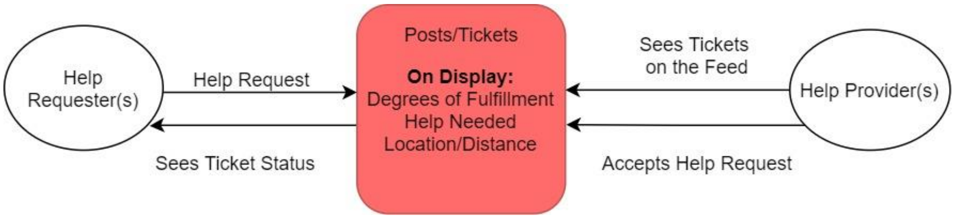
- Request location
- Help Requester username
- Picture(optional)
- Text (Max 150 characters) (Optional)
- Help Type List (With numeric quantity needed.)

Commented [DLMP4]: Add the missing bullet point on the on display section of the Posts/Tickets.

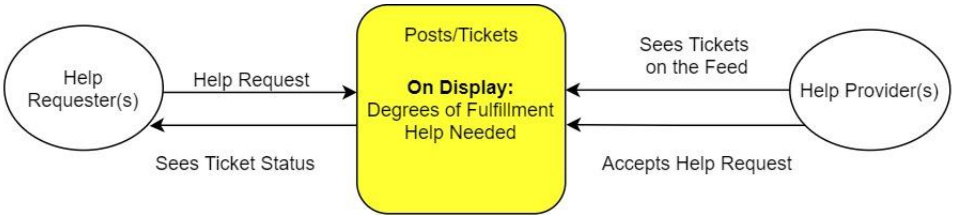
Tickets will be color coded. Each color representing the state/ degree of fulfillment of the ticket.:

Commented [DLMP5]: Missing a bubble at the top representing neutral state.

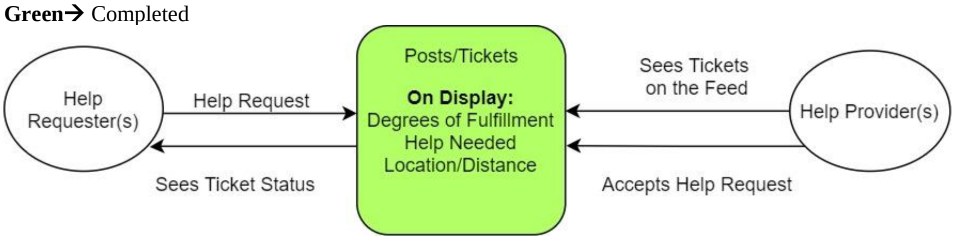
Red → Ticket request has not yet been accepted.

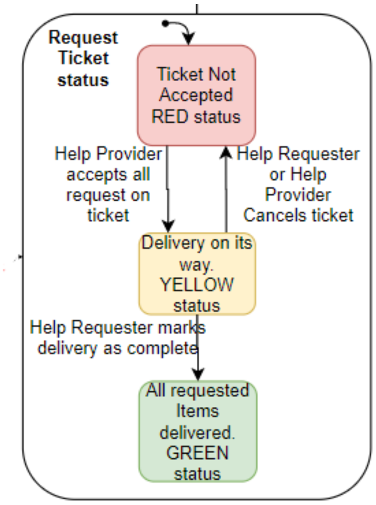


Yellow→ In progress. Help provider can deliver all items



Green→ Completed




Request Ticket status
Ticket Not Accepted RED status
Help Provider accepts all request on ticket
Help Requester or Help Provider Cancels ticket
Delivery on its way. YELLOW status
Help Requester marks delivery as complete
All requested Items delivered. GREEN status

Help Provider Ticket→ A ticket is like a social media post. It will contain the following:

Commented [d6]: Work on this!!! There are two types of ticket now!!! Provider Ticket and requester ticket.

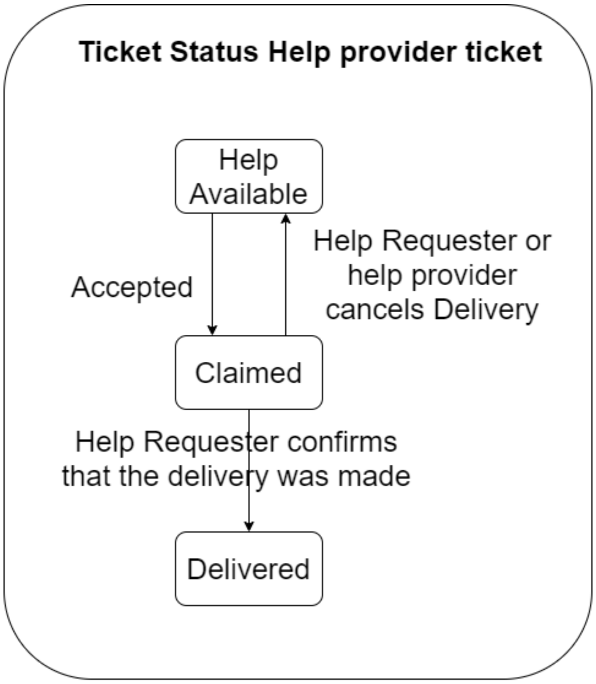
- Help provider location
- Help provider username
- Picture(optional)
- Text (Max 150 characters) (Optional)
- Help Type List (With numeric quantity needed.)
- Ticket status/ Degrees of fulfillment

Commented [DLMP7]: Add the missing bullet point on the on display section of the Posts/Tickets.

Help provider tickets have 3 degrees of fulfillment:

- Available
- Claimed but not yet delivered.
- Delivered

Commented [d8]: Add some sort of visual indicator of status


Ticket Status Help provider ticket
Help Available
Accepted
Help Requester or help provider cancels Delivery
Claimed
Help Requester confirms that the delivery was made
Delivered

Ticket feed/timeline→

- All created tickets will be displayed on the feed. Users can filter the tickets by level of fulfilment, location and/ or help items.

Commented [d9]: Requirment: Ticket filter

Commented [GS10R9]: ahora seran requests, que son el merge de posts y tickets pero con un ctrl+f estamos chilling no tienes que bregar con eso ahora

Creating a ticket→ A user can create a ticket at any time by posting it on a help provider feed or a help request feed.

Accepting a ticket→ Once a ticket has been selected by a user, they can accept or reject it. If accepted Ticket status changes.

Commented [d11]: This needs more detail.

Commented [GS12R11]: no creo que tengamos lo de messeging mano, por ahora dejalo asi sencillo pq no hemos llegao that far

Our system/Application can be express in two views:

- **Ticket feed views→**
  - Help provider view.
  -
- **User profile view→** Displays all of user information:
  - Username
  - List of posted tickets with their status.
  - List of accepted tickets and updates in status changes.

Commented [d13]: Should tickets have a unique indicator for request or provide.

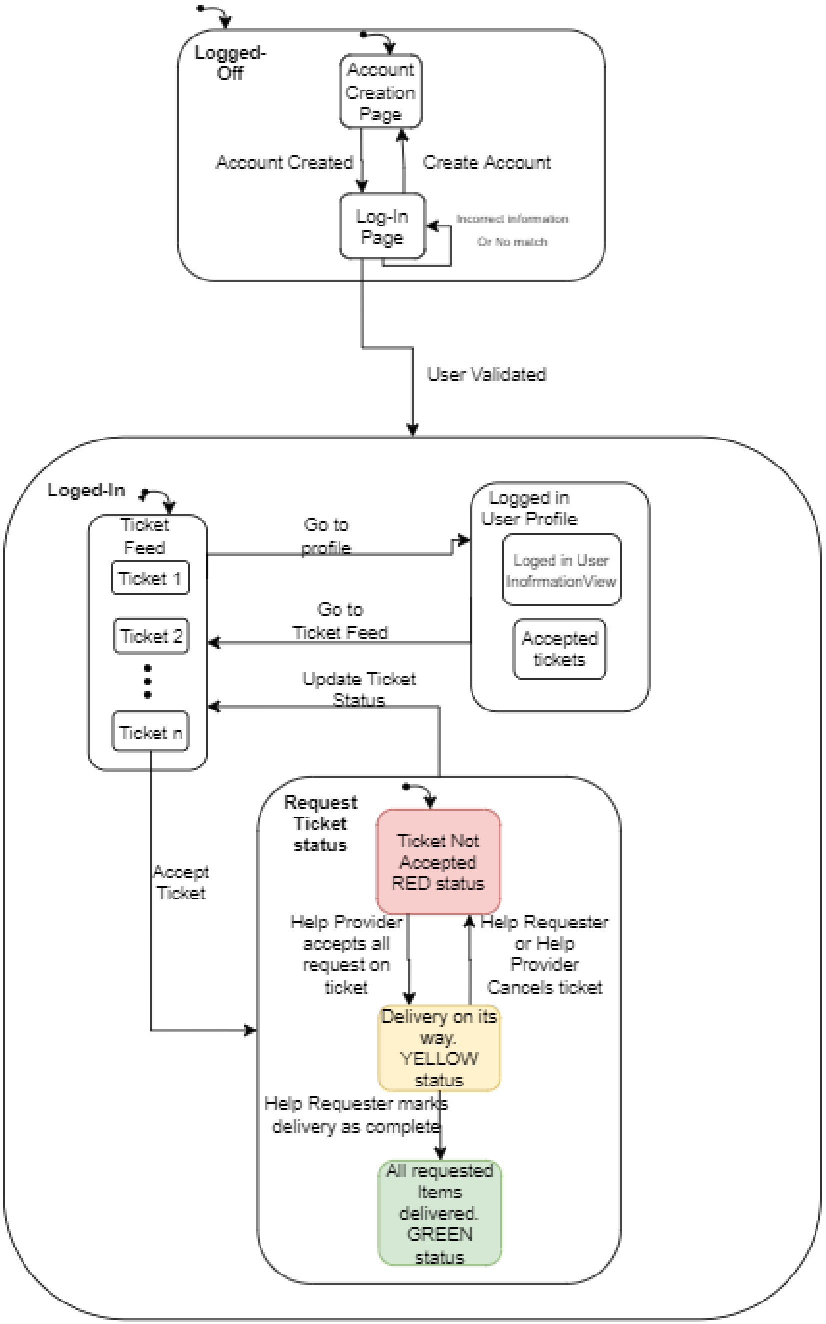
Commented [GS14R13]: no entiendo tu pregunta, what I can say es que posts are limited to requests now

Commented [DP15R13]: Perfect. I will change it now

Ticket/Post Confirmation System:

Communication Between Users

Direct communication between users is established outside of the system. Users will provide contact information like email, and phone number to establish communication with one another.


Logged-
Off
Account
Creation
Page
Account Created
Create Account
Log-In
Page
Incorrect information
Or No match
User Validated
Loged-In
Ticket
Feed
Ticket 1
Ticket 2
Ticket n
Go to
profile
Logged in
User Profile
Loged in User
Inofrmation View
Accepted
tickets
Go to
Ticket Feed
Update Ticket
Status
Accept
Ticket
Request
Ticket
status
Ticket Not
Accepted
RED status
Help Provider
accepts all
request on
ticket
Help Requester
or Help
Provider
Cancels ticket
Delivery on its
way.
YELLOW
status
Help Requester marks
delivery as complete
All requested
Items
delivered.
GREEN
status

Requirement	**Test(s)**	**Test Level of Completion**	**Implementation** (YES, NO, Partial)
User Registration must ask for name, email.	**Unit Tests**	PASS	YES
User Login	**Unit Tests** Tests for input formats in login page **Acceptance Tests** Used selenium to verify end to end workflow. **Integration tests** End to end workflow handled.	PASS	YES
User Profile must contain user name, location, user rating, request post history, and help provider post history, and some contact information.	**Unit Tests** Tests for input formats, valid handling and data. **Integration tests** End to end workflow handled, missing front end JavaScript implementation.	Partial PASS	PARTIAL
User must be able to make a post.	**Unit Tests** Tests for input formats, valid handling and data. **Integration tests** End to end workflow handled, missing front end JavaScript implementation.	Partial PASS	YES
User must be able to update their location.	**Unit Tests** Tests for input formats, valid handling and data. **Integration tests** End to end workflow handled, missing front end JavaScript implementation.	Partial PASS	PARTIAL

Commented [d16]: Not sure what things needs to be included here…

Commented [GS17R16]: yees

User must be able to delete an existing post.	**Unit Tests** Tests for input formats, valid handling and data. **Integration tests** End to end workflow handled, missing front end JavaScript implementation.	Partial PASS	PARTIAL
User can edit an existing post.	**Unit Tests** Tests for input formats and valid information. **Integration tests** End to end workflow handled, missing front end JavaScript implementation.	Partial PASS	PARTIAL
Users must be able to filter tickets according to parameters of interest. **Parameters of interest**: Location, Ticket type, Help type, User Name, degrees of fulfillment, date of post.	**Unit Tests** Tests for input formats, valid handling and data. **Integration tests** End to end workflow handled, missing front end JavaScript implementation.	Partial PASS	PARTIAL
Help Request Tickets/Posts must show degrees of fulfillment. In other words, ticket status.	**Unit Tests** Tests for input formats, valid handling and data. **Integration tests** End to end workflow handled, missing front end JavaScript implementation.	Partial PASS	PARTIAL
Usability testing	**Questionnaire and Demo**	PASS	Partial

Commented [d18]: To what extent.

Commented [GS19R18]: por ahora no creo que logremos edit , pero si se logra implement seria description and title

Pre-test and PSSUQ (Post-Study System Usability Questionnaire)

PSSUQ follows a 7-point Likert Scale (+ NA option).

- Overall: the average scores of questions 1 to 16
- Sub-scale tests
    - System Usefulness (SYSUSE): the average scores of questions 1 to 6
    - Information Quality (INFOQUAL): the average scores of questions 7 to 12
    - Interface Quality (INTERQUAL): the average scores of questions 13 to 15

The sub-scales provide a more detailed breakdown of different factors affecting the website, software, system or product.

https://docs.google.com/forms/d/e/1FAIpQLSc2DJkt4_6jpIU4DFZ2j7HkjfjNaHh5TOfqZSlbd0vlpigaMA/viewform?usp=sf_link

Upon analyzing the questionnaire, we were able to pinpoint various weaknesses in our design, specifically for those who might not use the internet often and need help. Our original design mimicked various social media sites while maintaining a colorful user interface to soothe the experience. However, the application decided to drop most of its visual aspects for big and straight to the point posts in order to aid newcomers. Our intention was to maintain the application as open as possible, probably many senior citizens who want help or need help are not capable of handling the dynamic user interfaces presented in modern day sites so we opted for big bold texts to maintain the entry level low. The UI is currently under development, so we are continuously monitoring possible persona outcomes for the various changes made.

Unit Tests on User: All the methods in the User Handler file where validated, by comparing their output, with an expected result at hand. Output was also compared between methods themselves (methods which retrieve information), to make sure that consistency exists throughout the User Handler file. Other functions, like the Register and the Login function, where tested slightly differently. The register function was put to the test by first registering with all required fields, and then attempting to register with only some of the fields. The expected output, in which the latter is incorrect, was received. For the login function, a similar approach was taken, in which we tested that a user may not login with an incorrect password or an incorrect username, only with the correct choice of username and password.

Integration Tests on Routes: All routes created where verified by checking their respective status codes. The profile page was confirmed to not be able to be accessed unless a user is logged in. If an attempt is made to access it using routes, without being logged in, then you are redirected to the homepage. The register page was confirmed to be creating a new user once the forms were filled. The logout function was tested by verifying that the adequate status code was returned once logged out. It serves to verify and confirm proper backend handling as well as guaranteeing the front-end receives what is needed to implement the view.

Selenium Integration Tests on Front End: Selenium tests were used to simulate a user browsing and interacting through the front end's features. The simulated user attempts a login with information that is not stored in the database. It then proceeds to register some credentials and finally it proceeds to login to the profile section. This comprehensive test allows us to confirm that the front end, back end and database are all working adequately, in a straightforward manner. If the simulated user logs in correctly then, the posting feature is tested.