Functional Requirements

- System functionality to be performed and Information to be Processed:
    1. The application must prompt the user(volunteer) for the username, which is decided by the volunteers group. The volunteer group must share one password for all members, which is used for login
    2. After login, user(volunteer) shall be able to customize his/her profile
        1. User(volunteer) must be able to logout after successful login
    3. The application must allow user(passerby) direct access to the main functions of the app if the user indicates he/she is a passerby
    4. The application must request user(volunteer/passerby) for permission before accessing the user's location
        1. If user denies access, should user want to use any location enabled features within the application, he/she must provide a location manually
        2. If user grants access, all location enabled features should automatically retrieve the user's current location via GPS
    5. The application must request user(passerby) for permission before accessing the user's mobile phone camera and photo library
        1. If user denies access, any features which require camera access must display a warning to the user and further access to the feature will be denied
        2. If user grants access, any features which require camera access will automatically open up camera/photo library
    6. When the user(passerby) submits a photo of a stray cat, the photo must be analyzed to identify the breed of the cat through a pre-trained machine learning model
        1. The application must display all the cats of the same breed which currently exist in the database and must prompt the user to match the reported cat to one or none of the displayed stray cats
            1. If the user cannot find a match, the cat shall be considered a newly found stray cat
            2. If the user finds a match, the cat is an existing stray cat in the database
    7. When the cat exists in the database, the application must update the last seen location of the stray cat
    8. When the cat does not exist in the database, the application must prompt the user(passerby) to input details of the stray cat where applicable:
        1. Name
            1. Guidelines must be given to the user to ensure appropriate naming convention is adhered to
            2. Name must be text of at least 1 character and not more than 3 words
        2. Last seen location of stray cat
            1. If location is enabled, the last seen location of the stray cat must be automatically updated to be the current location of the user
    9. The application must display last-seen locations as well as brief descriptions of all the stray cats in the database when the user(volunteer) clicks on any of the map markers shown in the map, such as:
        1. The cat's name
        2. The cat's breed
        3. The cat's last location

10. When the user(passerby) files a report of an injured stray cat, the application must prompt the user to write a description of any visible injuries and upload a picture of said injuries
    1. The description must be text of greater than one word and not more than 120 words
    2. Guidelines must be given to ensure reliable picture taken: Picture taken of the injury must be clear, the injured body part must take up at least 50% of the frame
11. The application must display all reports of injured stray cats to aid volunteer groups in tending to them
    1. If user(passerby) does not want to send the injured stray cat to the vet:
        1. User(volunteer) must be notified of the last reported location of the injured cat
        2. If a particular case has been handled by a user(volunteer), it must automatically disappear from the list of injured stray cats
    2. If user(passerby) wants to send the injured stray cat to the vet:
        1. Nearest vet clinic function must be made available to the passerby
        2. Contact number and name of passerby must be taken in as input for the user(volunteer) to liaise with the user(passerby) directly
        3. Location of vet clinic the user(passerby) will be taking the injured stray cat to must be made known in a notification to users(volunteers) and a user(volunteer) is required to take over the case at the vet clinic
        4. After the case is taken over by a user(volunteer), it must automatically disappear from the list of injured stray cat
12. The application must allow the user(volunteer/passerby) to search for vet clinics by:
    1. Manual:
        1. Name
        2. Address (contains postal code)
        3. Office telephone number
        4. Users dragging the map manually
    2. Automatic:
        1. The application indicates the top 5 nearest vet clinic based on the user's location
13. The application must display detailed information about the vet clinics when the user(volunteer/passerby) clicks on any of the map markers shown in the map, such as:
    1. The vet clinic's full name
    2. The vet clinic's full address
    3. The vet clinic's telephone number
14. The application must show the most efficient pathway from the user's(volunteer/passerby) location to a vet clinic if the user indicates that they want to visit that particular vet clinic
15. The application must display detailed information about existing volunteer groups in Singapore to aid the user in enquiring for adoption, such as:
    1. The volunteer group's full name
    2. The volunteer group's full address
    3. The volunteer group's telephone number
    4. The volunteer group's email address

- Interface with other systems:

1. The application must be able to use GPS to track user's(volunteer/passerby) current location when submitting reports of stray cats
2. The application must be able to use Wi-fi to communicate all actions executed with Huawei cloud database
3. The application must be able to retrieve data on locations of licensed vet clinics in Singapore through data.gov.sg API
4. The application must be able to use Google Maps API to show locations of all cats in the database, unless adopted by the member of the public or taken care of by any volunteer group

Non-functional Requirements

1. Usability:
    1. Strive for consistency
        1. Use of consistent sequence of actions for similar situations
        2. Use of identical terminology for prompts, menus and help screens
    2. Offer informative and visual feedback
        1. Display messages when user performs certain actions (e.g. users will be prompted a message when submission of photo is successful)
        2. Display messages when certain process is on-going (e.g. retrieve data)
        3. Display error messages when invalid input is detected or certain process fails
        4. Push notifications to volunteers when a stray cat has been located by a passer-by
    3. Permit easy reversal of actions
        1. Confirmation of actions before proceeding with irreversible actions (e.g. "back" buttons when the user is submitting a photo of a stray cat at every page before official submission to the database)
    4. Reduce short term memory
        1. Use of simple displays with to-the-point descriptions for users to locate and identify stray cats in the database easily
    5. Prevent errors:
        1. Ensure that correct login information is input before granting user access
2. Reliability
    1. After refreshing the mobile app, the full system functionality should be restored
    2. All information displayed must be accurate as of last updated information from the database
3. Performance
    1. The application must be able to retrieve/upload data for not more than 20 seconds
    2. The application must be able to update location of the cats in the database after every photo submission via GPS
    3. The application must be able to notify any new injured cats to volunteers in real-time
4. Supportability
    1. The database will be deployed to a cloud platform, provided by Huawei
    2. The application can be downloaded on Android and iOS systems
5. Security
    1. The application will perform encryption on newly created passwords during registration
    2. During user login, the encrypted passwords stored in the database will be matched with the password provided as input by the user

3. Password field will be masked to protect user privacy