WhatsUp RAD

1 - Introduction

1.1 - Purpose of application

The WhatsUp application allows you to share important or fun information with your friends, by creating points on a global map. It's possible to comment and rate these points, and to sort them with different filters, such as distance from your current position.

1.2 - General characteristics of application

The applications consists of two major parts, a client and a server. The client is basically a map and some buttons, and other options to enable creation and viewing of annotations. The server saves all annotation data, and handles authentication.

1.3 - Scope of application

The application is limited to only create annotations in a certain way, and to view them. It has no connection with other Google features or social networks like Facebook.

1.4 - Objectives and success criteria of the project

- A map where it's possible to pan around and zoom.
- Add annotations.
- View, comment and rate annotations.
- To create annotations an account is required.

1.5 - Definitions, acronyms and abbreviations

Annotation: A point on the map (set of coordinates) with attached information such as a title, description, category, rating and a set of comments.

Annotation marker: An object containing the essential information from its corresponding annotation, enough to be displayed on a map or in a list. In practice, an annotation with only coordinates, title, ID and rating.

Reference point: A location used as a quick-jump point in the map view, and a distance reference in the list view. Reference points can be saved as favorites, and selected from the favorites list. The current location of the device (provided by GPS or equal) is treated as one reference point among the saved points, and can be selected in an equal manner.

Server: Remote system accessed over the Internet which maintains the user database for authentication purposes, provides the stored annotation information to any requesting device, and handles contributed information (annotations, comments, ratings) from authorized users.

The app: The front-end of the system, GUI and logic that the user can experience. The app is the first part of the system that is installed and run on the Android platform

Client: The application, usually in the context of communication with the server.

Content provider: The local database which correlates to a subset of the server database, concerning annotation/annotation marker information.

Service: Second part of the system on the Android platform. This is a self-containing process serving the content provider with updates from the server (REST communication), serving the app with relevant data from the content provider and/or server.

2 - Proposed application

2.1 - Overview

With the WhatsUp application one is able to create annotations on a map, and view annotations created by other users. It's possible to sort them according to distance to a selected position, comment and rate them. To create annotations one has to be logged in.

2.2 - Functional requirements

1 - Location-related Requirements

1.1 - View map

Scenario: View map
Trigger: Starting the app
Precondition: None
Basic path
The user starts the application. When the app launches, a map with nearby annotations will be shown.

Status on completion: The map activity and the list activity should be started and loaded, and the map activity should be shown

1.2 - Scroll map
Scenario: View map
Trigger: Scrolling or swiping the map
Precondition: The map activity is active
Basic path
When the user scrolls or swipes on the map the viewport of the map will be moved accordingly. When the map stops in one location, new annotations will be loaded and shown.
Exceptional path:
If no new annotations can be fetched due to any connection issues, see requirement
Status on completion: The view will be centered around a new location, specified by the user, and the annotations passing the current filters will be shown.

1.3 - Zoom map
Scenario: View map
Trigger: Pinching the map
Precondition: The map activity is active
Basic path
When the user zooms in or out the map the zoom level will be changed. New annotations will be loaded and shown when the user is not zooming any longer.
Status on completion: The view will be centered around a new location, specified by the user, and the annotations passing the current filters will be shown.

1.6 - Filtering by category
Scenario: View map
Trigger: Choosing to filter the map
Precondition: None
Basic path
The user is presented with a list of categories. The user selects one or several of these categories, and chooses to view the map again.
Post condition
The map will now be filtered and only show annotations that matches at least one of the categories.

1.7 - Stop filtering by category
Scenario: View map
Trigger: Choosing to filter the map
Precondition: None
Basic path
The user is presented with a list of categories and an alternative to reset the filter. The user chooses to reset the filtering.
Post condition
Now all types of annotations will be shown.

1.8 - View annotation
Scenario: View annotation
Trigger: Choosing to view an annotation
Precondition: The map activity is active

Basic path
An annotation with title, rating, a description and comments is shown to the user.

1.9 - Close annotation
Scenario: View annotation
Trigger: Choosing to stop viewing an annotation
Precondition: An annotation is being viewed
Basic path
When choosing to go back from/close an annotation the user returned to whatever activity was open before.
Status on completion: The last viewed main activity is shown, and the application has discarded the closed annotation.

1.10 - Open list view of annotations
Scenario: View list of nearby annotations
Trigger: Choosing to list nearby annotations
Precondition: The map activity is active
Basic path
When choosing to list annotations nearby, annotations will be shown to the user, with the closest annotations on top. The user will be shown the title, rating and distance to the annotation.

1.11 - Close list view of annotations
Scenario: View list of nearby annotations
Trigger: Choosing to stop viewing the list of annotations
Precondition: The list view is active
Basic path
When choosing to go back from/close the list view the user is returned to whatever activity was open before.

1.14 - View favorite reference positions
Scenario: View map
Trigger: Choosing to view favorite
Precondition: Map or list view active
Basic path
A list of favorites and the current location is shown.
Status on completion
The list of favorites is shown.

1.15 - Remove favorite reference positions
Scenario: View map.
Trigger: Choosing to remove a favorite.
Precondition: List of favorites is being viewed.
Basic path
A confirmation dialog is shown to the user. The user is shown a notification after accepting the confirmation.

Status on completion
The favorite reference is never shown to the user again.

1.16 - Set reference point of map

Scenario: View map
Trigger: Choosing the reference point of the map.
Precondition: List of favorites is being viewed
Basic path
The user chooses the reference point either from the favorites or from the current location.
Status on completion
The center of the map is moved to the new reference point and information about the distance to annotations will be shown relative to the reference point.

1.17 - Load reference point of the map

Scenario: View map
Trigger: Starting the app
Basic path
The last active reference point is loaded.
Status on completion
The last reference point, or if no such could be found, a new one created as specified in requirement 1.18 is active when starting the app

1.18 - Creation of new reference point of the map

Scenario: View map
Trigger: Unable to find earlier version of map reference point.
Basic path
The application tries to calculate the phone's current location by GPS. If this fails, it tries 3G. If this fails, it tries wifi. If all these fall through, it attempts to use the IP of the phone to guess which country the user is in, and puts the reference point somewhere central in that country.
Status on completion
A map reference point should be loaded for the application, ready to be showed on the map, and used for range-from calculations.

1.19 - Save reference point of the map

Scenario: View map (some other scenario?)
Trigger: Closing the app
Basic path
The active reference point should always be stored when the app is closed.
Status on completion
The last reference point is saved in a persistent state, ready to be loaded when next starting the app.

1.20 - Return to reference point of map

Scenario: View map
Trigger: Choosing to return to the active reference point on the map.
Precondition: A reference point has been picked
Basic path
The user chooses to return to the reference point.
Status on completion
The map is centered upon the reference point.

1.23 - Starting the database

Scenario: Starting application
Trigger: Starting the application
Precondition: None
Basic path:
When the application starts, it should create a new database locally, unless one is already present in a healthy condition from earlier runs.
Exceptional path:
If the database is unable to be started, a message will be displayed to the user, and the application will continue running normally, but with no local caching, only network requests.

1.24 - Local caching of data

Scenario: Storing data locally
Trigger: The application retrieves data from the server
Precondition: None
Basic path:
When the application gets requested data from the server, the content provider updates the database with the new data, unless it's the same as .
System state on completion: The database should be updated to reflect the new information from the server.

1.25 - Run network calls in separate thread

Scenario: All activities that use the network.
Trigger: Initiating a network call.
Precondition: None
Basic path:
Network calls will be started in a separate thread to ensure that the UI doesn't freeze.
Exceptional path:

1.26 - Run network calls in a separate service

Scenario: All activities that use the network.
Trigger: Initiating a network call.
Precondition: None
Basic path:
To ensure network operations can finnish and update the content provider even if incoming calls interrupts the activity or the user exits the application data management will run in a service that can run without activities open.
Exceptional path:

1.27 - Close the service when its inactive

Scenario: Using the app
Trigger: The service has no more pending tasks
Precondition: None
Basic path:
The service dies.

1.28 - Application should run without network

Scenario: Doing a task that needs network connection without having network connection.
Trigger: The user or the application initiates a network task.

Precondition: The network is not enabled.
Basic path:
The application should not produce errors when attempting a network operation while no network is available

2 - Contribution-related Requirements

2.1 - Create annotation

Scenario: Add annotation
Trigger: The user chooses to create an annotation.
Precondition: User is logged in
Basic path:
The user selects what point on the map to add the annotation to. Then the user enters the title, category and description of the annotation. When finished the user submits the annotation. A confirmation of the submission is shown. Title is required, while description may be blank.
Exceptional path:
The user is not logged in/registered. The log in/register activity is shown. See 3.1 and 3.2.
Exceptional path:
User fails to provide a valid title. Upon submission, the activity will stay without sending any data, and show highlighting on the erroneous entry to alert the user.
Status on completion: The map or list view activity is shown depending on what activity was open before creating the annotation.

2.2 - Rate annotation

Scenario: Rate annotation
Trigger: The user rates an annotation.
Precondition: User is viewing an annotation written by another user, user is logged in
Basic path:
After rating the new score of the annotation is shown to indicate success.
Exceptional path:
The user is not logged in/registered. The log in/register activity is shown. See 3.1 and 3.2.
Status on completion: The rating is added to the database in the backend and the annotation view is still active.

2.3 - Change rating

Scenario: Rate annotation
Trigger: The user rates an already rated annotation.
Precondition: The user has already rated the annotation and is logged in (else its not possible to decide if the user has rated)..
Basic path:
The changed rating is displayed to the user.
Status on completion: The map or list view activity is shown depending on what activity was open before creating the annotation. The same user cannot rate an annotation twice, so if the same rating was given, nothing should change. If the new rating is not equal to the earlier one, the earlier one is removed and replaced by the new one.

2.4 - Comment an annotation

Scenario: Comment annotation
Trigger: The user chooses to comment annotation.
Precondition: User is logged in
Basic path:

The user enters the comment and confirms. The new comment is displayed at the top of the comment view.
Exceptional path:
The user is not logged in/registered. The log in/register activity is shown. See 3.1 and 3.2.
Status on completion: The map or list view activity is shown depending on what activity was open before creating the annotation.

2.5 - Load more comments when scrolling

Scenario: Add annotation
Trigger: The user chooses to create an annotation.
Precondition: None
Basic path:
The user selects what point on the map to add the annotation to. Then the user enters the title, category and body of the annotation. When finished the user submits the annotation. A confirmation of the submission is shown.
Exceptional path:
The user is not logged in/registered. The log in/register activity is shown. See 3.1 and 3.2.
Status on completion: The map or list view activity is shown depending on what activity was open before creating the annotation.

2.6 - Sync annotation to keep data fresh

Scenario: Viewing annotation
Trigger: The user opens an annotation that has been changed remotely.
Precondition: There already exists annotations, that is changed from somewhere other than the local machine.
Basic path:
The user selects an annotation to view. The application polls the server, asking if the remote annotation differs from the locally cached value. If it has, or if there is no cached local value (i.e. this annotation has not been opened yet) the application pulls data for that annotation from the server.
Exceptional path:
If the connection to the server is lost, see requirement 1.22
Status on completion: The annotation is displayed with up to date values, as test case 1.8

2.7 - Information sent to server

Scenario: Add annotation / Comment on annotation / Rate annotation
Basic path: No information is sent to the server until the user confirms the contribution. When the

2.8 - Possibility to change data during a contribution process

Scenario: Add annotation
Basic path: Any contribution process that takes several steps through multiple activities must provide an option to go back to a previous step and let the user change whatever data was submitted in that step, or even back out completely from the contribution process. User signals this by using the back button, the previous step shows and is ready for new input.
Exceptional path:
The user submits data on the final step. This is considered a final decision and the data is stored (sent to server or local database), all activities related to the particular contribution are closed.

State on completion:
The user backs out of a step in the contribution process and finds the previous step or the parent activity starting the process.

3 - User-related Requirements

3.1 - Register

Scenario: Register
Trigger: User tries to use a contribution feature
Precondition: The user is not logged in to an account.
Basic path:
From the log in activity the user chooses to create an account. The user enters desired user name, email and a password. When the user submits the form a message about a confirmation mail is shown. The user clicks the link in the mail and can then proceed to logging in.
Exceptional path:
The user enters invalid information such as an occupied user name, an email already in the database or an incorrectly formatted email address.
Status on completion: The user is returned to the activity the registration was initiated from.

3.2 - Log in

Scenario: Log in
Trigger: User tries to use a contribution feature
Precondition: User is not logged in, User has an account.
Basic path:
User is at the log in activity and provides user name and password. The information is sent to the server by “Log in” (button or equal). Server responds that the user is authenticated and the log in activity returns to the previous activity with the active user session.
Exceptional path 1:
Server responds “user not authenticated” due to no match for user name and password in account database. Log in interface should stay visible and indicate that the provided combination of user name and password does not exist in the account database. User may try basic path again.
Exceptional path 2:
User chooses return instead of log in. Log in activity withdraws and returns no active user session to the previous activity.

3.3 - Log out

Scenario: Log out
Trigger: User chooses log out (menu or button)
Precondition: User is logged in
Basic path:
The user session is terminated (account details no longer stored on the phone) and the user is notified when the process is done.
State on completion: There is no user session, and no password information is stored in the app. Last used user name information is stored for convenient log in.

3.4 - Automatic log in

Scenario: Log in
Trigger: Starting the app

Precondition: User has logged in before exiting the app. User has not logged out before exiting the app
Basic path:
When starting the app the user will be logged in without enering credentials.
State on completion: The user has an authenticated session and can use contribution features.

3.4 - Opt out of automatic log in

Scenario: Log in
Trigger: choosing to turn of automatic log in
Precondition: user has an account.
Basic path:
The user will not be automatically logged in when starting the app.
State on completion:
The user will not be automatically logged in when starting the app.

2.3 - Non-functional requirements

2.3.1 - Usability

1.4 - Uncluttered map

Scenario: View map
Precondition: None
Basic path
To make sure that the map is usable there should not be too many points on the map simultaneously and they should not be too close to each other. This will be achieved by having an upper limit for how many annotations can be shown and a minimal distance between the annotations shown. How this will be done is specified in 1.5.

1.5 - Showing relevant annotations

Scenario: View map
Precondition: None
Basic path
Make sure that when having to make a selection of which annotations are shown those shown will be chosen by their rating, how fresh they are and whether the user has commented/rated them.

1.12 - Indicate category/type of annotation with marker

Scenario: View map
Trigger: Choosing to view map
Precondition: None
Basic path
The markers of different annotations should have an icon that reflects their primary category.

1.21 - Show user when data is transmitted between the client and the server

Scenario: All related to communicating with the server.
Trigger: Any communication with the server.
Precondition: None
Basic path

To make sure that the user knows when data is sent and received over the network, some sort of indicator will always be present when that happens.
Status on completion
The indicator disappears/is not shown in an active state.

1.23 - Always show the current reference point at the top of the list

Scenario: Viewing the reference point list.

Trigger: Bringing up the reference point activity

Precondition: None

Basic path

The list of reference points always shows the current one at the top.

2.3.2 - Reliability

1.22 - Connection lost or corrupted

Scenario: Failure to communicate
Trigger: A request or expected sending from the system fails
Precondition: None
Basic path: If any communication fails, the server should discard any incomplete information received. The application should indicate for the user that the connection is lost or corrupted, and no data can be sent at the moment.
Status on completion: An error message that prompts the user to try again later should be shown.

2.3.3 - Performance

Network traffic should, with a good connection, not take too much time. The map view should be fluid and smooth.

2.3.4 - Supportability

The application is divided into smaller modules, to increase mod-ability.

2.3.5 - Implementation

The client is developed with Android SDK 2.2 including the Google API 2.2. The server runs the PHP CMS Drupal with the Services and Location modules to enable REST-communicatin and storage of geodata.

The site is available at whatsup.placebo.nu.

2.3.6 - Verification

Every night the latest version of the client is fetched from the repository, built and tested with unit tests. http://placebo.nu:8080/job/WhatsUp/ A coverage report is also generated. The most recent coverage report is available at whatsup.placebo.nu/coverage/coverage.html.

2.3.7 - Packaging and installation

It'll be packaged as an Android .apk file that easily can be installed with any Android phone.

2.3.8 - Legal

The application is released under the GNU GPLv3 license. Some Apache 2.0 licenced code with preserved copyright notices has been adapted and integrated into the project. These files

are available in the nu.placebo.whatsup.android.os and nu.placebo.whatsup.balloon packages with comments explaining the adaptations made.

2.4 - Test cases

1 - Location-related tests

1.1 - Launch application, view map

Test case description: Launch application, view map
Precondition: The application is installed and ready to run.
Test steps

1. Launch the application.
2. Verify that the map is displayed and nearby annotations are visible.

Related requirement: 1.1

1.2 - Scroll map

Test case description: Scroll map
Precondition: The application is running and the map view is active. (Test case 1.1 successful).
Test steps

1. Scroll the map to the sides by usual android scrolling (swiping the screen in the opposite direction the map should go).
2. Verify that the map moves in the correct direction.
3. Verify that new annotations has been loaded.
4. Verify that the shown annotations correspond to the filter values set.

Related requirement: 1.2

1.3 - Zoom in and out

Test case description: Zoom in and out
Precondition: The application is running and the map view is active. (Test case 1.1 successful).
Test steps

1. Zoom the map in and out, in the usual android zoom style (using two fingers and swipe them from each other, and to each other, respectively).
2. Verify that the map zooms in and out correctly.
3. Verify that new annotations has been loaded.
4. Verify that the shown annotations correspond to the filter values set.

Related requirement: 1.3

1.4.1 - Add category filters

Test case description: Add category filters.
Precondition: The application is running and the map view is active. (Test case 1.1 successful).
Test steps

1. Open the filter settings.
2. Choose one or more categories.
3. Return to the map view.
4. Verify that the selected filter is applied, by checking that only annotations that match at least one of the conditions are displayed.

Related requirement: 1.6

1.4.2 - Remove category filters

Test case description: Remove category filters.
Precondition: The application is running and the map view is active. (Test case 1.1 successful).
Test steps

1. Open the filter settings.
2. Return filter settings to default.
3. Return to the map view.
4. Verify that all annotations are displayed.

Requirement: 1.7

1.5.1 - View annotation details

Test case description: View annotation details.
Precondition: The application is running and the map view is active. (Test case 1.1 successful).
Test steps

1. Click on an annotation.
2. Verify that the title, rating, description and comments are displayed.

Requirement: 1.8

1.5.2 - Close annotation details

Test case description: Close annotation details.
Precondition: An annotation is being viewed. (Test case 1.5.1).
Test steps

1. Close the annotation.
2. Verify that you return to the latest activity.

Requirement: 1.9

1.6.1 - View list of annotations

Test case description: View list of annotations.
Precondition: The application is running and the map view is active. (Test case 1.1 successful).
Test steps

1. Switch to the list view of closest annotations.
2. The annotations should be sorted with the closest at the top, with title and distance showing.

Requirement: 1.10

1.6.2 - Switch from list view to map view

Test case description: Close list view of annotations.
Precondition: The list view of annotations is active (test case 1.6.1).
Test steps

1. Switch back to map view.
2. Map is shown.

Requirement: 1.11

1.8.1 - Add a reference point

Test case description: Add a reference position.

Precondition: The reference point view is active.
Test steps

1. Click add reference position.
2. A map is shown.
3. Choose a location for the reference position.
4. Enter title.
5. Save.
6. The position is saved, and the application returns to the list of reference points including the new point.

Requirement: 1.13

1.8.2 - View reference points

Test case description: View reference points.
Precondition: The map view or the list view of annotations is active. At least one favorite exists (test case 1.8).
Test steps

1. Start the “view reference points” activity.
2. A list of reference points and the GPS position is shown.

Requirement: 1.14

1.8.3 - Remove favorite reference point

Test case description: Remove favorite reference point.
Precondition: The list view of favorite positions is active (test case 1.8.2).
Test steps

1. Remove a favorite.
2. A confirmation is displayed.
3. After confirmation, the favorite is removed.

Requirement: 1.15

1.9 - Set active reference point

Test case description: Set the reference position of the map.
Precondition: The list view of favorite positions is active (test case 1.8.2).
Test steps

1. Set the reference with either a favorite from the list, or the gps position.
2. The center of the map is moved to the new position, and the distance to annotations is calculated relatively to the new reference position.

Requirement: 1.16

1.10 - Load reference point during start up

Test case description: Load reference point during start up.
Precondition: The application is installed and ready to run.
Test steps

1. Start the application.
2. During start up the last active reference point should be loaded.
3. If no previously created reference point exist, a new point is created with the phone’s current location (requirement 1.18).

Requirement: 1.17

1.11 - Default reference point

Test case description: Load a default reference point when the reference point list is emptt
Precondition: The application is installed and ready to run. No previously created reference point should exist, like in test case 1.8.1.
Test steps

1. Start the application.
2. Check that there is a reference point called “My Location”. (which is the GPS position where applicable)

Requirement: 1.18

1.12 - Save created reference points at shutdown

Test case description: Created reference points are saved when the app is closed.
Precondition: The application is running and a reference point is created and active.
Test steps

1. Restart the application.
2. Verify that the reference point is loaded and active.

Requirement: 1.19

1.13 - Return to active reference point

Test case description: Return to active reference point.
Precondition: The application is running and a reference point is created and active.
Test steps

1. Return to reference point.
2. The map is centered around the reference point.

Requirement: 1.20

1.14 - Indicate server traffic

Test case description: Indicate server traffic with an icon or graphical effect.
Precondition: None
Test steps

1. Perform test case 2.1.1.
2. During creation of the annotation, verify that the traffic to the server is indicated with a status indicator.

Requirement: 1.21

1.15 - Connection lost

Test case description: Give response when connection is lost.
Precondition: None
Test steps

1. Perform test case 2.1.1.
2. During creation of the annotation, disconnect the phone and verify that an error message is displayed.

Requirement: 1.22

1.16 - Starting the database

Test case description: Start the database when application is started, or verify that one is already present in a healthy condition since an earlier run.
Precondition: Create several reference points and choose one as current reference point that is not “My location”, then close the application.
Test steps

1. Start the application again and go to reference points.
2. Verify that the points you had when you closed are still in the list, and the current reference point is still selected.

Requirement: 1.23

1.17 - Caching annotation locally

Test case description: The database caches annotations locally.
Precondition: Test case 2.1.1 should have been successfully ran.
Test steps

- Close network connections and attempt to load the annotation created.
- If it shows, the database is working in this aspect.

Requirements: 1.24

1.18 - Current reference point

Test case description: Showing the current reference point at the top of the list
Precondition: None
Test steps

1. From the main view (the map), click on the cross-hair button in the menu to open the reference point.
2. Make sure the current reference point (the one with a marker to it's left) is on the top of the list.

Requirements: 1.23

1.19 - Run offline

Test case description: The application should not produce errors when running in offline and attempting network operations. Only telling the user that these kind of tasks requires network connection.
Precondition: None
Test steps
Disconnect the device and attempt to refresh the map view.
Requirements: 1.24

2 - Contribution related test cases

2.1.1 - Create annotation

Test case description: Create annotation
Precondition: User is logged in and showing the map view
Test steps

1. Choose to add annotation
2. Check that the map is decorated with a cross-hair and that no annotation are shown (the point selection map).
3. Scroll the map so that it is centered on the desired annotation point.
4. Choose done/add/next/whatever.
5. Check that the point selection map disappears, and the annotation details form appears.
6. Enter title and annotation.
7. Submit the form.
8. Verify that the map shown is the standard map and not the point selection map.
9. Verify that a success notification is shown.

Related requirement: 2.1

2.1.2 - Test validation of create annotation form

Test case description: Create annotation
Precondition: User is logged in
Test steps

1. Choose Add annotation
2. Check that the map decorated with a crosshair, and without annotations is shown (point selection map).
3. Choose a location.
4. Choose done/add.
5. Check that the point selection map disappears, and the annotation details form appears.
6. Leave the title fields empty.
7. Submit the form.
8. Verify that and error message is shown.
9. Verify that the crea te annotation form still is shown.
10. Verify that the title field is highlighted as erroneous.

Related requirement: 2.1

2.1.3 - Change position during creation of an annotation

Test case description: Create annotation
Precondition: User is logged in
Test steps

1. Choose Add annotation
2. Check that the point selection map is shown
3. Aim for a map location
4. Click “done”
5. Check that the map disappears and the annotation details form appears
6. Click the back button
7. Make sure the map reappears for new point selection

Related requirement: 2.1, 2.8

2.2 - Comment on annotation

Test case description: Comment on annotation
Precondition: There is an existing annotation, user is logged in.
Test steps:

1. Choose an annotation to view (test case 1.5.1).
2. The annotation view should now be open with information about the chosen annotation; click the add comment button.
3. Write a comment in the comment box. The comment should be a non-empty string.
4. Click add button.
5. The view should now be back to the annotation, with the new comment appended to the comment list.

Related requirement: 2.4

2.3.1 - Rate an annotation

Test case description: Rate an annotation
Precondition: There is an existing annotation created by another user which has the focus and user is logged in
Test steps

1. Take note of the rating value.

2. Rate the annotation up.
3. Verify that the rating value in the annotation view has been increased.
4. Proceed with TC 2.3.3

Related requirement: 2.2

2.3.2 - Rate your own annotation

Test case description: Rate your own annotation
Precondition: There is an annotation created by the user(test case 2.1) which has the focus and user is logged in
Test steps

1. Verify that rate cannot be accessed

Related requirement: 2.2

2.3.3 - Rate an annotation you have already rated

Test case description: Rate an annotation you have already rated
Precondition: Test case 2.3.1 performed
Test steps

1. Make sure you still know the original rating value, and the current rating value.
2. Try rating up again
3. The rating should not change.
4. Rate down
5. Compare the current rating to the original rating, make sure it has decreased.

Related requirement: 2.2

2.4 - Contribute while not logged in

Test case description: Contribute while not logged in
Precondition: There is an existing annotation, user is not logged in
Test steps:

1. Touch to show the details of the annotation
2. At each of the following steps: Note that the log in activity shows up, and go back
3. Try rating the annotation
4. Try commenting on the annotation
5. Go back to the map view and try creating an annotation

Related requirement: 2.1, 2.2, 2.4

3 - User related test cases

3.1.1 - Register a new account

Test case description: Register a new account
Precondition: User not logged in
Test steps

1. Enter registration dialog by choosing to register or by trying to use a contribution feature.
2. Enter an email address and user name and pick a a password.
3. Submit the form. Verify that a success message is shown and that the user is prompted to confirm account by reading the mail.
4. Open mail and confirm account by clicking link in mail.

Related requirement: 3.1

3.1.2 - Registering with existing credentials

Test case description: Registering with existing credentials

Precondition: User not logged in
Test steps

1. Enter registration dialog by choosing to register or by trying to use a contribution feature.
2. Enter an already registered user name and a correct email and password.
3. Verify that an error message is shown when submitting the form.
4. Enter an already registered mail address by a unique user name and a correct password.
5. Verify that an error message is shown when submitting the form.

Related requirement: 3.1

3.2.1 - Log in with valid credentials

Test case description: Log in with valid credentials
Precondition: User not logged in
Test steps

1. Enter log in dialog by choosing to log in or trying to use a contribution feature.
2. Enter a valid user name and password.
3. Submit the form.
4. Verify that a success message is shown.
5. Verify that the option to log in is not available any longer.
6. Verify that the view now active is whatever the user was doing before logging in.

Related requirement: 3.2

3.2.2 - Log in with invalid credentials

Test case description: Log in with invalid credentials
Precondition: User not logged in
Test steps

1. Enter log in dialog
2. Enter an invalid (not registered) user name.
3. Verify that an error message is shown when submitting the form.
4. Enter a valid user name with an invalid password.
5. Verify that an error message is shown when submitting the form.
6. Verify that the option to log in is still available when browsing other views of the app.

Related requirement: 3.2

3.3 - Log out

Test case description: Log out
Precondition: User logged in
Test steps

1. Choose to log out.
2. Verify that a notice of success is shown.
3. Verify that the option to log in is now available.

Related requirement: 3.3

3.4 - Automatic log in

Test case description: Test automatic log in
Precondition: User logged in
Test steps

1. Make sure automatic log in is activated
2. Exit the app
3. Start the app

4. Verify that the user account has been automatically logged in,

Related requirement: 3.4

3.5 - Opt out of automatic log in

Test case description: Opt out of automatic log in
Precondition: User logged in
Test steps

1. Opt out of automatic log in
2. Exit the app
3. Start the app
4. Verify that the user has not been logged in automatically.

Related requirement: 3.5

2.5 - Planned future directions

- Adding and watching comments.
- Rating annotations and showing it on the annotation.
- New server algorithm for deciding which annotations to deliver at any specific zoom level based on physical pixel proximity to other annotations.
- Filters for the list of annotations, to sort it according to more things besides the range from the current reference position.

2.6 - Possible future directions

- Automatically get annotations when zooming and panning on the map.
- Use social networks e.g. Facebook to authenticate.

2.7 - References

N/A

Appendix

N/A