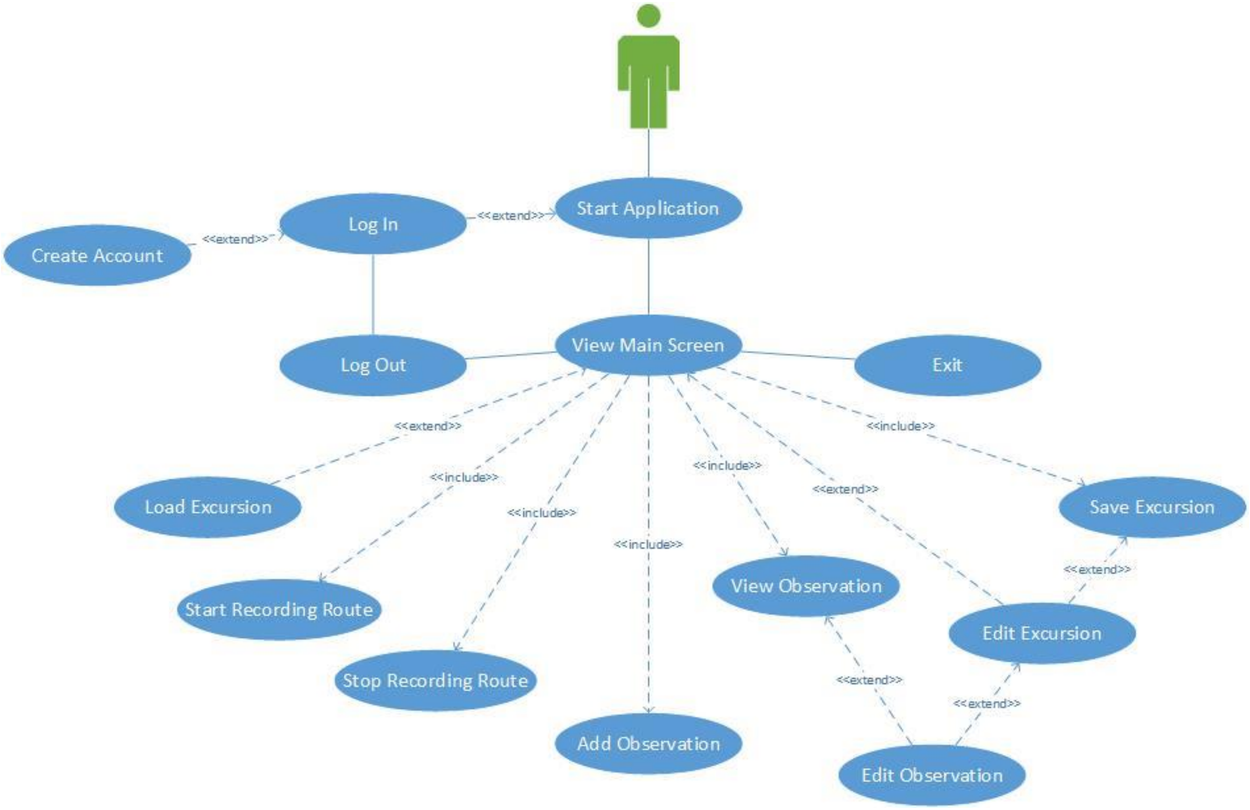
Follow Me Use Case Model



I. Use Case Name: **Start Application***

II. Participating Actors: User

III. Entry Conditions: User has downloaded and installed the application on an Android device. The user presses the application launch icon.

IV. Flow of Events:

   a. A splash screen, that includes the name of the application and the application logo, displays briefly on the screen.

   b. If the user is not already logged in, the application invokes the *Login* use case. Otherwise the View Main Screen use case is invoked.

V. Exit Conditions: The user is logged in and the main screen is loaded.

VI. Exceptions: The user was not able to log in.

VII. Quality Requirements:

I. Use Case Name: **Login***

II. Participating Actors: User

III. Entry Conditions: The user is not currently logged in.

IV. Flow of Events:

   a. The login screen is displayed that includes the following controls.
      - Login name and password fields
      - A button labeled Login
      - A button labeled Create Account

   b. If the user clicks the Create Account button they invoke the *Create Account* use case.

   c. Otherwise the user enters their login name and password and clicks the Login button.

   d. The application compares the credentials with the credentials stored locally on the device.

      i. If the user name is invalid, or the password is not authenticated, the fields are cleared and the flow of evens continues at step a.

      ii. Otherwise the application invokes the *View Main Screen* use case.

V. Exit Condition:

   a. User is logged in to the application and the main application screen is displayed.

VI. Exceptions:

   a. The user doesn’t supply valid credentials.

   b. The local database server is corrupted, preventing the credentials from being validated.

VII. Quality Requirements: The local database server works properly.

I. Use Case Name: **Create Account***

II. Participating Actors: User

III. Entry Condition: The user is not be logged into the application.

IV. Flow of Events:

a. A screen is displayed with the following controls.

- Login name field
- Email address field
- First and last name fields
- A password field and a confirm password field
- A button labeled Create Account.

b. The user enters their credentials.

a. If the two passwords don’t match an error message appears and both password fields are cleared.

c. The user presses the Create Account button.

b. The credentials are sent to the application server.

c. The application server checks that the email address and user name are not already registered. If either is already registered the field is cleared and an error message is displayed.

d. The application server stores the user information.

e. The application also stores the user information locally and logs the user in.

V. Exit conditions: The user has a valid account and is logged into the application.

VI. Exceptions:

a. The user doesn’t supply valid credentials.

b. The connection to the application server can not be established or is severed before the account is created.

VII. Quality Requirements:

a. The user’s email address must be a properly formatted email address.

b. The login name can only contain upper and lower case letters and numerals. It must also be at least 3 characters in length.

c. The password can only contain upper and lower case alphabet characters and numerals. It must also be at least 8 characters in length, start with an alphabet character and include at least one number.

d. The application must maintain a stable connection to the application server.

I. Use Case Name: **View Main Screen***

II. Participating Actors: User

III. Entry Condition: The user is logged into the application.

IV. Flow of Events:

   a. The users GPS location is regularly computed.

   b. If a current excursion is not loaded, the current excursion is set to a default excursion with no route points and no observations. A map of the area, containing the user's GPS location, is downloaded and displayed on the screen. The user's GPS location is also plotted on the map.

   c. If the current excursion is a saved excursion, its route and observations are displayed on the map. If the users location is within the viewing area, the location is marked on the map.

   d. A menu bar is displayed at the top of the screen with the following buttons.

      - Load Excursion
      - Start Recording Route
      - Add Observation
      - Stop Recording Route
      - Edit Excursion
      - Save Excursion
      - Log Out
      - Exit

   e. The buttons that are active depends on the included use cases that are being invoked.

      i. The Add Observation, Edit Excursion, Save Excursion, Log Out, and Exit buttons and are always active.

      ii. The Load Excursion button and the Start Recording Route button is active if the application is not recording the user's location.

      iii. The Stop Recording Route button is active if the application is recording the user's location.

   f. When any button is pressed a corresponding use case is invoked.

   g. If an observation mark is displayed on the map and the user presses the observation the View Observation use case is invoked.

V. Exit Conditions: The user presses the Log out or Exit buttons.

VI. Exceptions:

VII. Quality Requirements: The device must have GPS capabilities.

I. Use Case Name: **Load Excursion***

II. Participating Actors: User

III. Entry Condition: User has the main screen loaded.

IV. Flow of Events:

   a. If the current excursion has changes that have not been saved, the application asks the user if he wants to save the changes.
   b. If the device has an Internet connection, the application updates a list of publically available (on the device and on the Internet) excursions.
   c. The application displays a scrollable list of available excursions. The application also displays a button labeled Cancel.
   d. The user clicks an excursion that he wants to view.
   e. If the excursion data is not on the device, it is downloaded onto the device.
   f. The application's current excursion is set to the new excursion.
   g. The route of the excursion is plotted on the map.
   h. The locations of the observations are marked on the map.

V. Exit conditions: The application's current excursion is set to a new excursion and the route and observations points of the excursion are plotted on the screen or the user presses cancel and the previous excursion is still loaded.

VI. Exceptions:

   a. The user selects a route requiring downloading, but the application fails to download it and a message is displayed on the screen.

VII. Quality Requirements:

   a. Device must have sufficient permanent storage to store the downloaded excursions.
   b. An Internet connection is needed to download excursions.

I. Use Case Name: **Start Recording Route***

II. Participating Actors: User

III. Entry Condition: The application is displaying the main screen and the application is not recording the GPS location of the user.

IV. Flow of Events:

   a. At a regular interval the application calculates the user's GPS location, stores the location and plots the location on the map.

V. Exit Conditions: The user's location is continually updated in the local database and on the map.

VI. Exceptions:

VII. Quality Requirements: The device must have a device that can calculate the GPS location of the device.

I. Use Case Name: **Add Observation***

II. Participating Actors: User

III. Entry Condition: The application is displaying the main screen.

IV. Flow of Events:

    a. A screen is displayed with the following controls.
        - Title field
        - Description field
        - GPS location field
        - A button labeled Cancel
        - A button labeled Commit

    b. The application calculates the current GPS location and sets the GPS location field.

    c. The user may edit all fields and when complete hits the Commit button if he wants the observation saved in memory.

V. Exit Conditions: A new observation is saved in memory and a mark for the observation is added on the map.

VI. Exceptions: The user presses the Cancel button.

VII. Quality Requirements:

I. Use Case Name: **View Observation***

II. Participating Actors: User

III. Entry Condition: The application is displaying the main screen and the user clicked on a observation mark on the map.

IV. Flow of Events:

    a. A scrollable box is displayed on the screen that contains the name, GPS location, and description of the observation along with a button labeled Back.

    b. If the user created the excursion an Edit button also appears on the screen. If the user presses edit, the Edit Observation is invoked.

    c. When the user if finished reading about the observation, he presses the Back button which causes the View Main Screen use case to be invoked.

V. Exit Conditions: The user viewed the observation.

VI. Exceptions:

VII. Quality Requirements:

VIII. Use Case Name: **Edit Observation***

IX. Participating Actors: User

X. Entry Condition: The user chose to edit an observation either from the View Observation use case or the Edit Excursion use case.

XI. Flow of Events:

  a. A screen is displayed with the following controls.
    - Title field
    - Description field
    - GPS location field
    - A button labeled Cancel
    - A button labeled Commit

  b. The user may edit all fields and when complete hits the Commit button if he wants the observation saved in memory.

XII. Exit Conditions: The user viewed the observation.

XIII. Exceptions: The user pressed Cancel.

XIV. Quality Requirements:

I. Use Case Name: **Stop Recording Route***

II. Participating Actors: User

III. Entry Condition: The application is displaying the main screen and the application is recording the user's GPS location.

IV. Flow of Events:

  a. The application stops recording the user's GPS location.

V. Exit Conditions: The application is not recording the user's GPS location.

VI. Exceptions:

VII. Quality Requirements:

I. Use Case Name: **Edit Excursion***

II. Participating Actors: User

III. Entry Condition: The application is displaying the main screen.

IV. Flow of Events:

  a. A screen is displayed with the following preference controls.
    - Excursion Name field
    - Excursion Description field
    - Toggle controls to determine if the route is for walking or driving
    - Toggle controls to determine if the route is private or public
    - A list of observations titles along with an edit and a delete buttons next to each observation.
    - A button labeled Cancel
    - A button labeled Commit

  b. If the name of the excursion is the same as the name of the default excursion an asterisk appears to the right of the name field.

  c. The user may edit the name and description and select the toggles.

  d. If the user presses the edit button next to an observation the Edit Observation use case is invoked.

  e. If the user presses the delete button next to an observation the observation is deleted from the excursion.

  f. If the user presses cancel, the changes are not saved and the View Main Screen use case is invoked.

  g. If the user presses commit, the application checks to see if the excursion name is different from the default excursion name. If not, a message appears to the right of the excursion name and the changes are not made. Otherwise, the changes are saved and the View Main Screen use case is invoked.

V. Exit conditions: Modifications to the current excursion’s properties are saved or the user presses cancel. The View Main Screen use case is invoked.

VI. Exceptions:

VII. Quality Requirements:

I. Use Case Name: **Save Excursion***
II. Participating Actors: User
III. Entry Condition: The application is displaying the main screen and some element of data for the excursion has changed.
IV. Flow of Events:
    a. If the name of the excursion is the default excursion name the Edit Excursions use case is invoked with the Cancel button on the Edit screen made inactive.
    b. The excursion data is saved to the local database.
    c. If an Internet connection exists, the excursion is uploaded to the application server and a message indicating the excursion was saved and uploaded is briefly displayed on the screen. Otherwise a message indicating that the excursion was saved but not uploaded is displayed.
V. Exit Conditions: The excursion is saved to the local database.
VI. Exceptions:
VII. Quality Requirements: An Internet connection is required in order to upload the excursion to the application server

I. Use Case Name: **Log Out***
II. Participating Actors: User
III. Entry Condition: The application is displaying the main screen.
IV. Flow of Events:
    a. The application displays a dialog box and asks the user if they are sure that they want to log out.
    b. If the user selects *Yes*, the user is logged off and the Login use case is invoked.
    c. Otherwise, the View Main Screen use case is invoked.
V. Exit Conditions: The user is logged off and the log in screen is displayed.
VI. Exceptions: The user decided he does not want to log out and selects *No* when the dialog box is displayed.
VII. Quality Requirements:

I. Use Case Name: **Exit***

II. Participating Actors: User

III. Entry Condition: The application is displaying the main screen.

IV. Flow of Events:

   a. The application checks if an excursion is being uploaded to the application server. If so, the application displays a message to inform the user that an excursion is being loaded and waits for the upload to complete.

   b. When uploading is complete, the application terminates.

V. Exit Conditions: The application terminates.

VI. Exceptions:

VII. Quality Requirements: