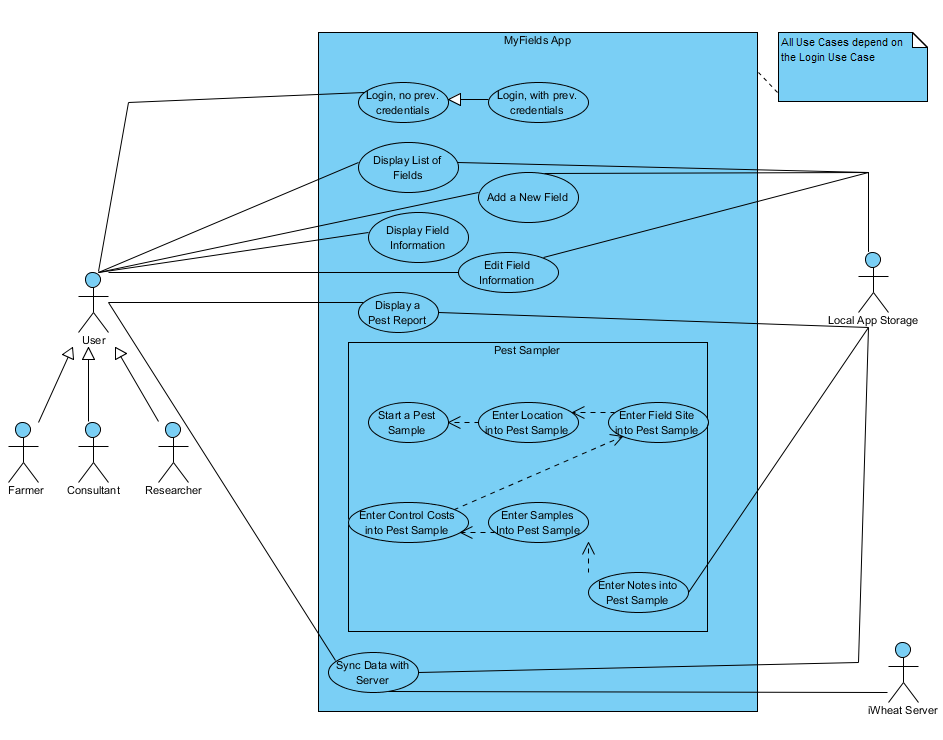
# Requirements

## Use Case Diagram



## Use Case Descriptions

### 1.2.1. Login, no previous credentials

**Actors**: User

**Stakeholders and Needs**:

User – Login to system to interact.

**Preconditions**: User has started system.

**Postconditions**: Fields List is displayed.

**Trigger**: User starts system.

**Basic Flow**:

1. System displays form with inputs for username and password, and login button
2. User inputs username and password
3. User clicks “Login” button
4. System authenticates with iWheat.org to retrieve user’s fields and associated info
5. System stores retrieved info and displays it in Fields List page

**Extensions**:

4a. If user credentials are invalid, return to step 1 and display “Invalid username/password.”

### 1.2.2. Login, with previous credentials

**Actors**: User

**Stakeholders and Needs**:

User – Login to system to interact.

**Preconditions**: User has started system, and has logged in to system previously.

**Postconditions**: Fields List is displayed.

**Trigger**: User starts system.

**Basic Flow**:

1. System retrieves credentials from secure local storage
2. System authenticates with iWheat.org to retrieve user’s fields and associated info
3. System checks retrieved info against locally stored info
4. System updates local storage to the most recent info and displays Fields List page

**Extensions**:

2a. If user credentials are invalid, go to step 1 of use case 1.2.1.

4a. If information does not match, fields have been updated while app was closed, via the website. Website information should take precedence.

### 1.2.3. Display List of Fields

**Actors**: User

**Stakeholders and Needs**:

User – View list of all fields, scroll through list

**Preconditions**: User is logged in to the system

**Postconditions**: User sees list of fields displayed

**Trigger**: User clicks “Login” with valid credentials OR valid credentials have been detected from a previous run

**Basic Flow**:

1. System uses user credentials to retrieve user’s field information from the website.
2. System parses retrieved information into Fields objects.
3. System checks local storage for Fields objects.
4. System builds and displays a form with each of those Fields objects to the user.

**Extensions**:

3a. If local fields objects are detected, and they do not match, the more recent object should be written into local storage.

### 1.2.4. Add a New Field

**Actors**: User

**Stakeholders and Needs**:

User – Add a new field to the list of user’s fields

**Preconditions**: User is logged in to the system

**Postconditions**: User has a new field displayed in their list of fields.

**Trigger**: User clicks on the “Add Field…” button on the Fields List page, or “Add a Field” from the Pest Sampler “Select a Field” page.

**Basic Flow**:

1. System builds and displays a form asking for each piece of information necessary to build a new field, with buttons for “Save” and “Cancel”
2. User inputs required information, and any optional information.
3. User clicks “Save” or “Cancel.”
   1. If User clicks Save, a new Field object is created from the entered information
   2. If User clicks Cancel, “Are you sure?” is displayed in a message box with “Yes” or “No” options.
      1. If “Yes” is pressed, information is discarded and Fields List is displayed
      2. If “No” is pressed, User returns to previous form.
4. System builds and displays the updated form with each field, including the new, to the user.

**Extensions**:

1a. ABSOLUTE necessary information is currently: Field Name, Latitude, Longitude

1b. Additional optional information: Size (acres), Soil Type, Tillage System, Irrigation Type

3a. Field Name, Latitude, Longitude, and Size should be validated. Other values are dropdowns.

### 1.2.5. Display Field Information

**Actors**: User

**Stakeholders and Needs**:

User – View information related to a specific field.

**Preconditions**: User is logged in to the system

**Postconditions**: User sees all information related to a specific field

**Trigger**: User clicks on a specific field in the Fields List OR clicks a link to that field from a page related to that field.

**Basic Flow**:

1. System retrieves the Field object associated with the named field.
2. System builds and displays a form containing all information associated with that field, including previous pest samples.

**Extensions**:

2a. Information to Display: Field Name, Latitude, Longitude, Size (acres), Soil Type, Tillage System, Irrigation Type

### 1.2.6. Edit a Field

**Actors**: User

**Stakeholders and Needs**:

User – Edit information associated with a field.

**Preconditions**: User is logged in to the system, and is on the desired field’s information page

**Postconditions**: System updates field information.

**Trigger**: User clicks on the “Edit Field” Button on the Field Information page.

**Basic Flow**:

1. System modifies the Field Information form to have editable fields, and a “Save” and “Cancel” button.
2. User clicks on the desired area to edit, and inputs the desired information.
3. User clicks “Save” or “Cancel.”
   1. If User clicks Save, that field object is updated with the new information.
   2. If User clicks Cancel, “Are you sure?” is displayed in a message box with “Yes” or “No” options.
      1. If “Yes” is pressed, information is discarded and Fields Information page is displayed.
      2. If “No” is pressed, User returns to previous form.
4. System returns to the Field Information form, which is updated if necessary.

**Extensions**:

1a. Necessary information is currently: Field Name, Latitude, Longitude, Size (acres), Soil Type, Tillage System, Irrigation Type

3a. Field Name, Latitude, Longitude, and Size should be validated. Other values are dropdowns.

### 1.2.7. Start a Pest Sample

**Actors**: User

**Stakeholders and Needs**:

User – Create a pest sample for a field.

**Preconditions**: User is logged in to the system

**Postconditions**: User is taken to the Sampling Method page of the Pest Sampler wizard

**Trigger**: User clicks on Pest Sampler page in app.

**Basic Flow**:

1. System builds and displays a form asking for the Sampling Method (dropdown) input, with buttons to “Continue” or “Cancel”
2. User clicks the dropdown and clicks one of the values.
3. User clicks either “Continue” or “Cancel”
   1. If Continue is clicked, the system will create a new pest sample object to modify with further information.
   2. The system will then display the “Location” page of the Pest Sampler.
   3. If Cancel is clicked, the system will return the user to the Fields List page.

**Extensions**:

1a. Dropdown values: Glance N Go (Greenbug), Wheat Stem Sawfly

3a. At this point, the system should validate that the user has selected a value in the dropdown; if not, return to step 2 and display a message stating “Select a sampling method!”.

### 1.2.8. Enter a location into the Pest Sampler

**Actors**: User

**Stakeholders and Needs**:

User – Create a pest sample for a field.

**Preconditions**: User is logged in to the system, has completed the “Sampling Method” page, and clicked Continue.

**Postconditions**: User is taken to the Site page of the Pest Sampler wizard

**Trigger**: User clicks on Continue on the “Sampling Method” page.

**Basic Flow**:

1. System builds and displays a form showing the detected device location on a map.
2. User may wish to move locations; should be able to if so desired.
3. User clicks either “Continue,” “Cancel,” or “Back”
   1. If Continue is clicked, the system will add the centerpoint location on the map to the created Pest Sample.
   2. The system will then display the “Sample Site” page to the user.
   3. If “Back is clicked, the system will display the “Sampling Method” page to the user.
   4. If User clicks Cancel, “Are you sure?” is displayed in a message box with “Yes” or “No” options.
      1. If “Yes” is pressed, information is discarded and Fields List page is displayed.
      2. If “No” is pressed, User returns to previous form.

**Extensions**:

3a. At this point, the system should validate that the user has centered the map over a valid location; if not, return to step 2 and display a message “Select a valid location!”.

### 1.2.9. Enter a Field Site into the Pest Sampler

**Actors**: User

**Stakeholders and Needs**:

User – Create a pest sample for a field.

**Preconditions**: User is logged in to the system, has completed the “Sampling Method” page and “Location” page, and clicked Continue.

**Postconditions**: User is taken to the Costs page of the Pest Sampler wizard

**Trigger**: User clicks on Continue on the “Location” page.

**Basic Flow**:

1. System builds and displays a form showing a dropdown with a list of the user’s fields, or the option to add a new field, with a “Continue,” “Back,” and “Cancel” buttons.
2. User will input either a dropdown value or “Add a new field” checkbox
   1. If User selects to add a new field, the system will display a box showing the previously selected location and asking for the field name, with a “Create” button.
   2. The user will enter the field name, then select “Create”
3. User clicks either “Continue,” “Cancel,” or “Back”
   1. If Continue is clicked, the system will do one of the two following:
      1. Create a new field based on the inputted name and location.
         1. Link the current pest sample to the new field.
      2. Link the current pest sample to the selected field.
   2. The system will then display the “Costs” page to the user.
   3. If “Back is clicked, the system will display the “Location” page to the user.
   4. If User clicks Cancel, “Are you sure?” is displayed in a message box with “Yes” or “No” options.
      1. If “Yes” is pressed, information is discarded and Fields List page is displayed.
      2. If “No” is pressed, User returns to previous form.

**Extensions**:

3a. At this point, the system should validate that either a dropdown value has been selected or a new field entered; if not, return to step 2 and display a message stating “Select or create a field!”.

### 1.2.10. Enter Control Costs into the Pest Sampler.

**Actors**: User

**Stakeholders and Needs**:

User – Create a pest sample for a field.

**Preconditions**: User is logged in to the system, has completed the “Sampling Method” page, “Location” page, and “Field Site” page, and clicked Continue.

**Postconditions**: User is taken to the “Samples” page of the Pest Sampler wizard

**Trigger**: User clicks on Continue on the “Field Site” page.

**Basic Flow**:

1. System builds and displays a form showing two dropdown with headers for Control Cost and Crop Value, with a “Continue,” “Back,” and “Cancel” buttons.
2. User will click the Control Cost dropdown, and enter a value.
3. User will click the Crop Value dropdown, and enter a value.
4. User clicks either “Continue,” “Cancel,” or “Back”
   1. If Continue is clicked, the system will add the inputted control and crop value costs to the pest sample.
   2. The system will then display the “Samples” page to the user.
   3. If “Back is clicked, the system will display the “Field Site” page to the user.
   4. If User clicks Cancel, “Are you sure?” is displayed in a message box with “Yes” or “No” options.
      1. If “Yes” is pressed, information is discarded and Fields List page is displayed.
      2. If “No” is pressed, User returns to previous form.

**Extensions**:

1a. Valid values for Control Cost: 4-12, integer values

1b. Valid values for Crop Value: 2.5 – 7, incrementing by .5

4a. At this point, the system should validate that both dropdown values have been entered; if not, return to step 2 and display a message stating “Select both control cost and crop value!”.

### 1.2.11. Enter Samples into the Pest Sampler.

**Actors**: User

**Stakeholders and Needs**:

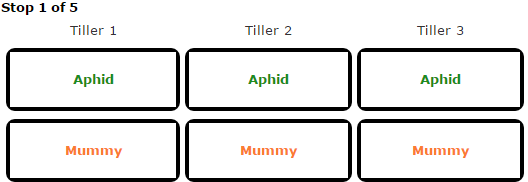
User – Create a pest sample for a field.

**Preconditions**: User is logged in to the system, has completed the “Sampling Method” page, “Location” page, “Field Site” page, and “Costs” page, and clicked Continue.

**Postconditions**: User is taken to the “Notes” page of the Pest Sampler wizard

**Trigger**: User clicks on Continue on the “Costs” page.

**Basic Flow**:

1. System builds and displays a form showing a table with checkboxes as the following:
   1.  with “Continue,” “Back,” and “Cancel” buttons.
2. User will select checkboxes on the table.
3. User clicks either “Continue,” “Cancel,” or “Back”
   1. If Continue is clicked, the system will add to the total number of detected Aphids and Mummys based on the checkboxes selected to the current sample object.
   2. The system will then display the next “Sample” page to the user, or Notes if 5 samples have been collected.
   3. If “Back is clicked, the system will display the “Costs” page to the user if it is the first sample, or the previous “Sample” page otherwise.
   4. If User clicks Cancel, “Are you sure?” is displayed in a message box with “Yes” or “No” options.
      1. If “Yes” is pressed, information is discarded and Fields List page is displayed.
      2. If “No” is pressed, User returns to previous form.

**Extensions**:

4a. At this point, the system should validate that the user has selected at least one checkbox; if not, display a message stating “Select at least one value!”.

### 1.2.12. Enter Notes into the Pest Sampler

**Actors**: User

**Stakeholders and Needs**:

User – Create a pest sample for a field.

**Preconditions**: User is logged in to the system, has completed the “Sampling Method” page, “Location” page, “Field Site” page, “Costs” page, and “Samples” pages, and clicked Continue.

**Postconditions**: User is taken to the “Pest Report” page of the Pest Sampler wizard

**Trigger**: User clicks on Continue on the last “Samples” page.

**Basic Flow**:

1. System builds and displays a form showing text entry boxes for Notes and Other Pests Observed with “Continue,” “Back,” and “Cancel” buttons.
2. User will either enter text into the boxes or not.
3. User clicks either “Continue,” “Cancel,” or “Back”
   1. If Continue is clicked, the system will add the desired notes and other observed pests to the current sample.
   2. The system will then display the “Pest Report” page to the user.
   3. If “Back is clicked, the system will display the last “Samples” page to the user.
   4. If User clicks Cancel, “Are you sure?” is displayed in a message box with “Yes” or “No” options.
      1. If “Yes” is pressed, information is discarded and Fields List page is displayed.
      2. If “No” is pressed, User returns to previous form.

**Extensions**:

3a. It is not required for the user to enter any text, so text should not be validated for notes.

### 1.2.13. Display a Pest Report

**Actors**: User

**Stakeholders and Needs**:

User – Review a pest sample associated with a field

**Preconditions**: User is logged in to the system, has completed all pages within the Pest Sampler and clicked “Continue,” OR clicks on a pest sample displayed on a Field Information page.

**Postconditions**: User is taken to the “Pest Report” page.

**Trigger**: User clicks on Continue on the “Notes” page of the Sampler, or clicks on a pest sample displayed on a Field Information page.

**Basic Flow**:

1. System builds and displays a form showing all information associated with a pest sample, with a button to “Edit” the sample.
2. If the user clicks “Edit,” all information becomes editable boxes the user may click and change, and a “Save” and “Cancel” button shall be displayed at the bottom of the page.
3. If the user clicks “Save”, all information shall be compared to the last known information for the sample, and if a mismatch occurs then the new information shall overwrite the old.
   1. If User clicks Cancel, “Are you sure?” is displayed in a message box with “Yes” or “No” options.
      1. If “Yes” is pressed, information is discarded and the Pest Report page is displayed.
      2. If “No” is pressed, User returns to previous form.

**Extensions**:

1a. All information includes: Location (map), Field Site, Sampling Method, Control Cost, Crop Value, Threshold for Treatment, Recommendation, Stop Data (stop #, Greenbug count, Mummy count), Notes, and Other Pests Observed.

### 1.2.14. Sync Data with Server

**Actors**: User

**Stakeholders and Needs**:

User – Store all local information with the iWheat server.

**Preconditions**: User is logged in to the system

**Postconditions**: All local information to the app is synchronized with the iWheat server’s information

**Trigger**: User selects the menu and clicks “Sync,” OR performed periodically.

**Basic Flow**:

1. System retrieves all field information for the user from the iWheat server, and constructs Fields objects with linked Pest Samples.
2. System compares the created objects; if a mismatch is detected, that field and associated pest samples are put in to a list for synchronization with the server.
   1. If a field is not in the list provided by the server, it will also be added to the sync list.
3. Once all fields have been compared, the system will serialize each field marked for sync to the iWheat server.
4. The server will store that data for the user for later usage and retrieval.

**Extensions**:

2a. If a field is provided by the server that is not detected by the app, that field should be added to the app’s local list.