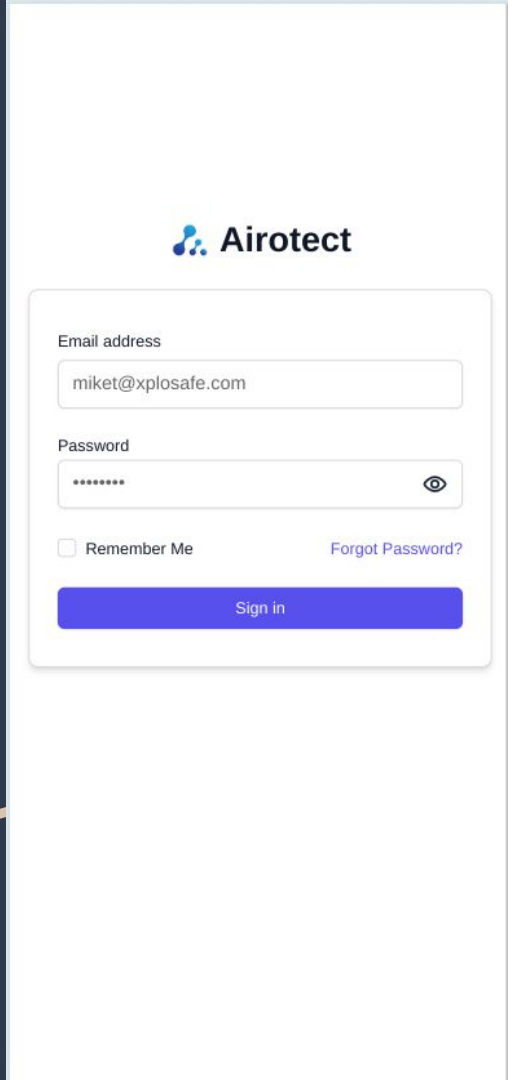
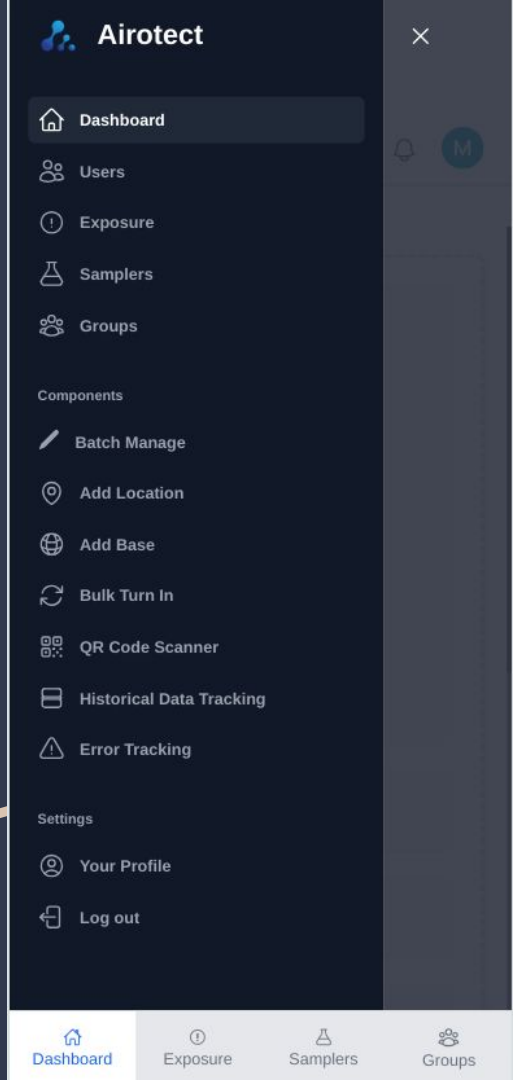
Airotect Mobile



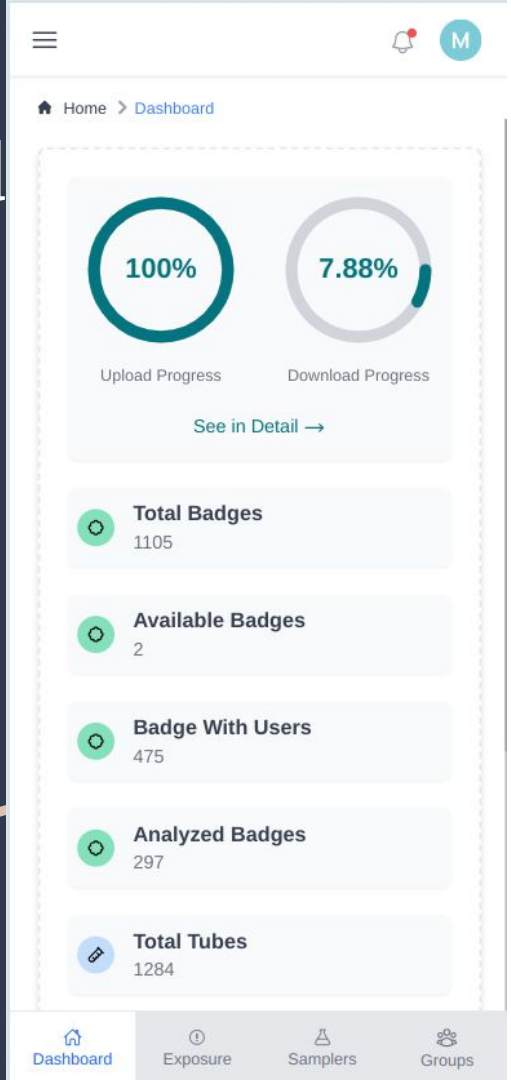
1. **Description**:
    a. **First-Time Online Login**: User enters credentials, the app authenticates via the server, and stores credentials in SQLite for offline use.
    b. **Offline Login**: On subsequent logins, the app checks the stored credentials in SQLite, allowing access without an internet connection.
2. **Key Points** :
    a. First-time login requires internet.
    b. Credentials securely saved in SQLite for offline access.
    c. Seamless experience both online and offline.
3. **Flowchart** :
    a. **User → Enter Credentials → [Online] Server Auth / [Offline] SQLite Check → Access Granted**

Menu Workflow



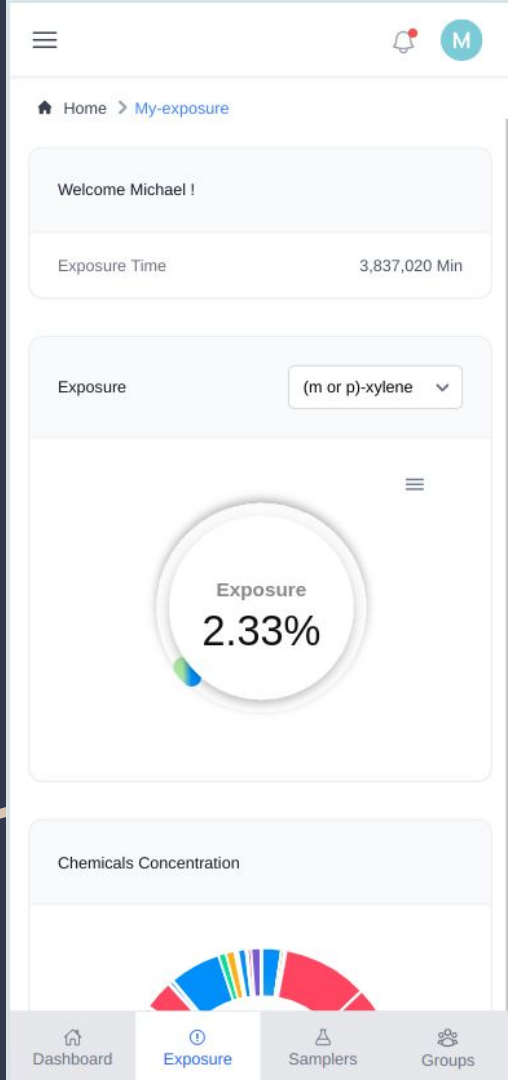
1. **Description:**
   a. **Menu Options Overview:** Provides easy access to a variety of key features including Dashboard, Users, Samplers, Groups, and components like Add Location, Add Base, Bulk Turn In, and more.
2. **Key Points:**
   a. **Dashboard:** Overview of key metrics and synchronization status.
   b. **Users:** View the user list and manage user accounts.
   c. **Samplers:** Access and manage the sampler list and sampling data.
   d. **Groups:** View and manage user groups.
   e. **Exposure:** Check user exposure data.
   f. **Batch Manage:** Update sampler batch numbers.
   g. **Bulk Turn In:** Download and upload files for bulk turn-in processes.
   h. **QR Code Scanner:** Scan QR codes to join groups.
   i. **Add Location/Add Base:** Components for adding locations and bases.
   j. **Historical Data Tracking:** Track total actions performed both online and offline.
   k. **Error Tracking:** Monitor errors encountered during synchronization from local SQLite to the server.
   l. **Profile:** View and edit user profile information.
   m. **Logout:** Log out of the application.
3. **Flowchart :**
   a. **User → Click Menu Icon → View Options → Select Option → Access Feature**

Dashboard



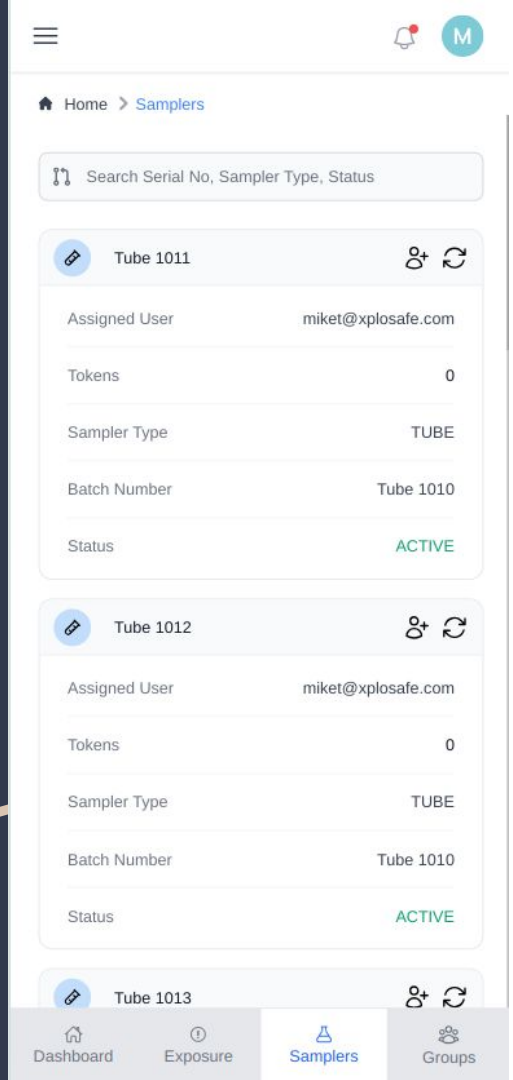
1. **Description**:
   a. After login, the **Dashboard** shows real-time data when the app is online.
   b. The dashboard displays:
      i. **Upload and Download Progress** as circular progress bars.
      ii. **Sampler Data**: Total Badges, Available Badges, Badges with Users, Analyzed Badges, and Total Tubes,Active Users,etc.
2. **Key Points**:
   a. **Online Mode**:
      i. Real-time data fetched from the server.
      ii. Data continuously updated (live synchronization).
      iii. New data from the server is stored locally in **SQLite** for future offline access.
   b. **Offline Mode**:
      i. Data displayed from the local **SQLite** database.
      ii. No live updates until the device reconnects to the internet.
      iii. Ensures seamless access to previously downloaded data.
3. **Flowchart**:
   a. **Online Workflow**:
      i. User logs in → Server data fetched → Data displayed in real-time → Data synced to SQLite.
   b. **Offline Workflow**:
      i. User logs in → SQLite data is loaded → Data displayed from local storage.

Exposure



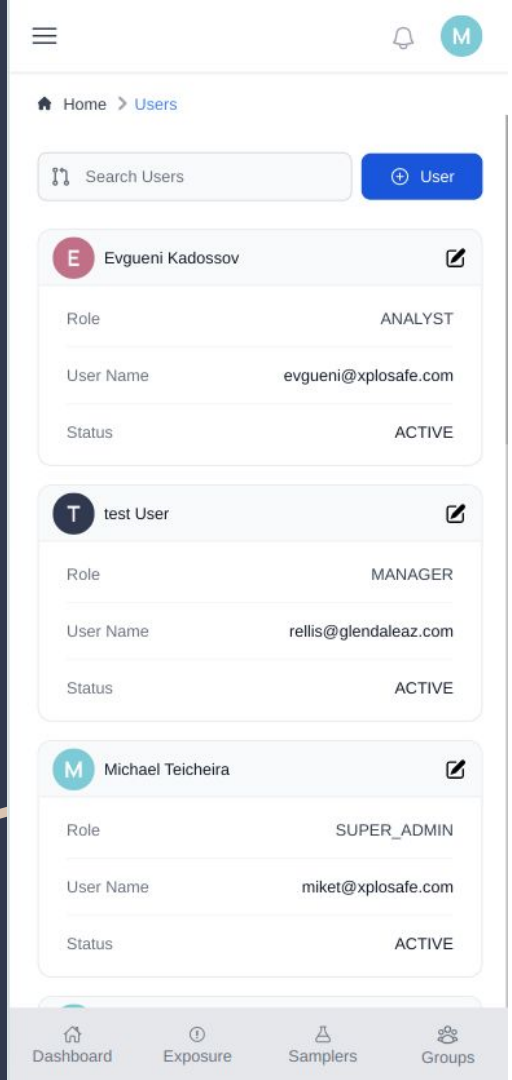
1. **Description**:
   a. On the **My Exposure** screen, the user can monitor their **Exposure Time** , **Most Found Chemical** and **Chemical Concentration**.
   b. Dropdown allows the user to switch between different chemicals (e.g., "m or p-xylene").
2. **Key Points**:
   a. **Online Mode**:
      i. Real-time exposure data is fetched from the server, providing the latest readings.
      ii. New exposure information is stored in **SQLite** for future offline use.
      iii. Users can switch between different chemicals for exposure data.
   b. **Offline Mode**:
      i. Exposure data is retrieved from the local **SQLite** database.
      ii. The data reflects the last synchronized readings when the device was online.
      iii. No live updates until the device reconnects to the internet.
3. **Flowchart**:
   a. **Online Workflow**:
      i. User logs in → Server data fetched → Data displayed in real-time → Data synced to SQLite.
   b. **Offline Workflow**:
      i. User logs in → SQLite data is loaded → Data displayed from local storage.

Samplers



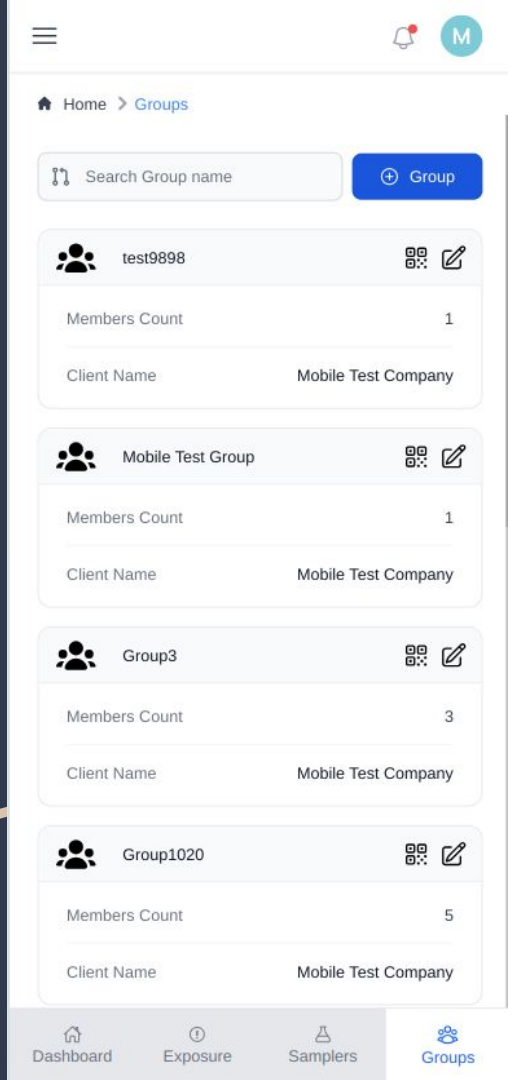
1. **Description**:
   a. The **Samplers Page** displays a list of all samplers, where users can:
      i. **Search** for specific samplers.
      ii. Navigate using **pagination**.
      iii. **Assign** or **Return** samplers as required.
   b. Supports both **online** and **offline** functionalities.
2. **Key Points**:
   a. **Online Mode**:
      i. Data is fetched in real-time from the **server**.
      ii. Users can assign or return samplers, and the data is updated on the server immediately.
      iii. New server data is stored locally in **SQLite** for offline access.
   b. **Offline Mode**:
      i. Data is displayed from the **SQLite** database (previously synchronized).
      ii. Users can **assign** or **return** samplers, and changes are stored locally in SQLite.
      iii. When the user reconnects online, all offline changes (assignments/returns) are **synchronized** with the server.
3. **Flowchart**:
   a. **Online Flow**:
      i. Navigate to Sampler → Data fetched from server → Sampler assigned/returned → Server updated and data stored in SQLite.
   b. **Offline Flow**:
      i. Navigate to Sampler → Data loaded from SQLite → Sampler assigned/returned → Changes stored in SQLite → Synced to the server when back online.

Users



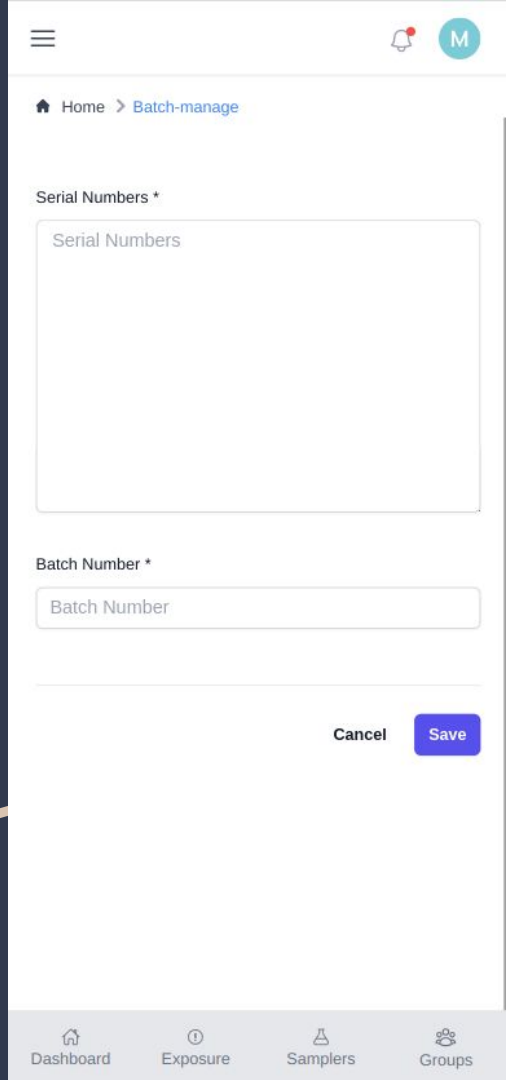
1. **Description**:
   a. The **Users Page** displays a list of all Users, where users can:
      i. **Search** for specific User.
      ii. Navigate using **pagination**.
      iii. **Create** and **Edit** Users as required.
   b. Supports both **online** and **offline** functionalities.
2. **Key Points**:
   a. **Online Mode**:
      i. Data is fetched in real-time from the **server**.
      ii. Users can Edit or Create the User, and the data is updated on the server immediately.
      iii. New server data is stored locally in **SQLite** for offline access.
   b. **Offline Mode**:
      i. Data is displayed from the **SQLite** database (previously synchronized).
      ii. Users can **Create** or **Edit** Users, and changes are stored locally in SQLite.
      iii. When the user reconnects online, all offline changes (edit/create) are **synchronized** with the server.
3. **Flowchart**:
   a. **Online Flow**:
      i. Navigate to User → Data fetched from server → User create/edit → Server updated and data stored in SQLite.
   b. **Offline Flow**:
      i. Navigate to User → Data loaded from SQLite → User create/edit → Changes stored in SQLite → Synced to the server when back online.

Groups



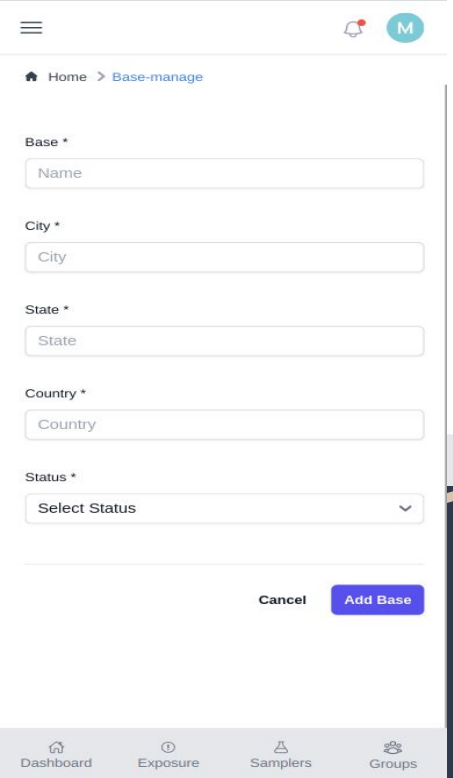
1. **Description**:
   a. The **Group Page** displays a list of all samplers, where users can:
      i. **Search** for specific Group.
      ii. Navigate using **pagination**.
      iii. **Create** and **Edit** Users as required.
      iv. Additionally Has Qr Code Scanner to Join that particular Group
   b. Supports both **online** and **offline** functionalities.
2. **Key Points**:
   a. **Online Mode**:
      i. Data is fetched in real-time from the **server**.
      ii. Users can Create or Edit the Group, and the data is updated on the server immediately.
      iii. New server data is stored locally in **SQLite** for offline access.
   b. **Offline Mode**:
      i. Data is displayed from the **SQLite** database (previously synchronized).
      ii. Users can **Create** or **Edit** Group, and changes are stored locally in SQLite.
      iii. When the user reconnects online, all offline changes (edit/create) are **synchronized** with the server.
3. **Flowchart**:
   a. **Online Flow**:
      i. Navigate to Group → Data fetched from server → Group create/edit → Server updated and data stored in SQLite.
   b. **Offline Flow**:
      i. Navigate to Group → Data loaded from SQLite → Group create/edit → Changes stored in SQLite → Synced to the server when back online.

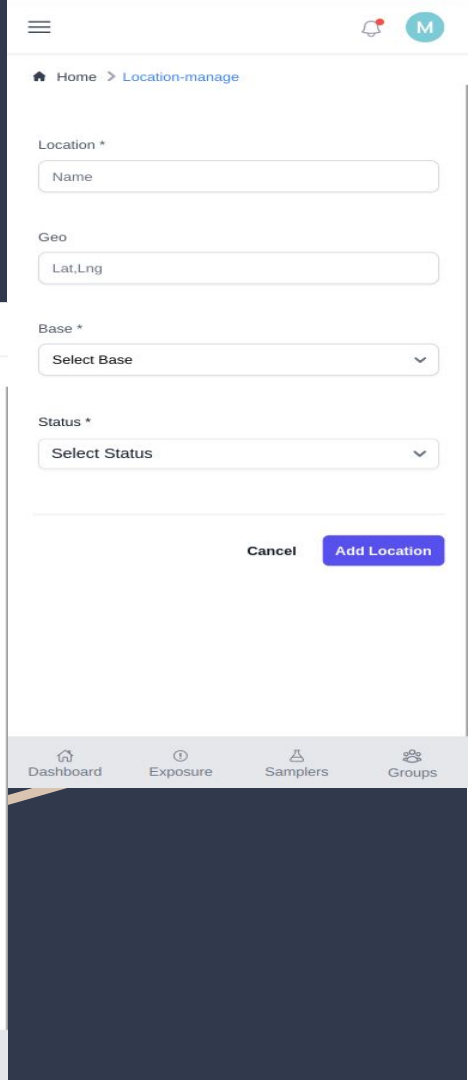
Batch Manage



1. **Description:**
    a. **Batch Manage Overview:** Allows users to enter a list of serial numbers and assign them to a batch number. Updates are handled differently based on connectivity status.
2. **Key Points:**
    a. **Enter Serial Numbers:** Users can input a list of serial numbers.
    b. **Assign Batch Number:** Enter the desired batch number for the samplers.
    c. **Online Update:** Directly updates all samplers with the specified batch number in real-time.
    d. **Offline Handling:** Stores the batch number and serial numbers in local SQLite.
    e. **Synchronization:** Updates all stored batch numbers to the server once the user reconnects online.
3. **Flowchart or Simple Diagram:**
    a. **User → Enter Serial Numbers and Batch Number**
    b. **Check Connectivity**
        i. **Online:**
            1. Directly Update Batch Numbers on Server
        ii. **Offline:**
            1. Store Data in Local SQLite
    c. **Reconnect Online:**
        i. Sync Batch Updates to Server

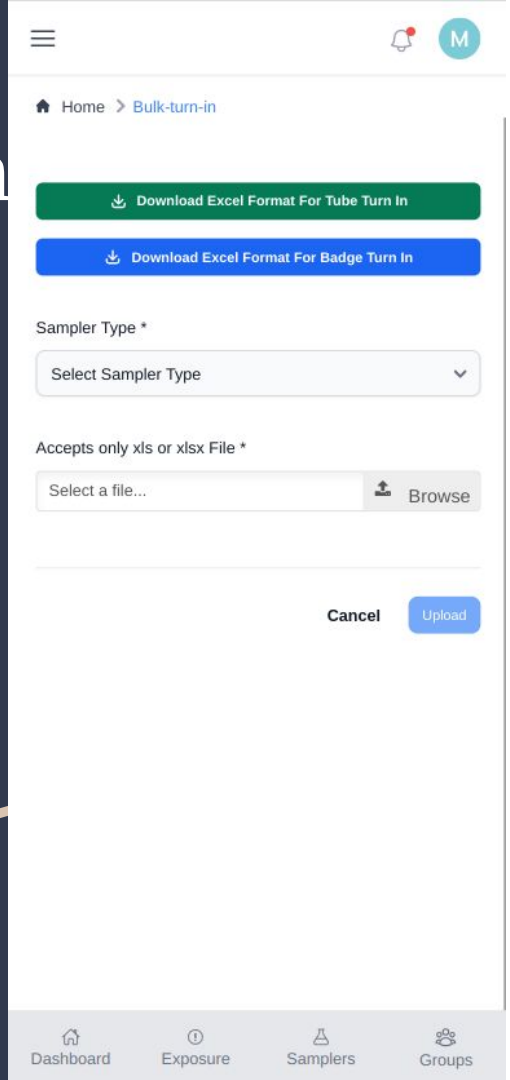
Location & Base





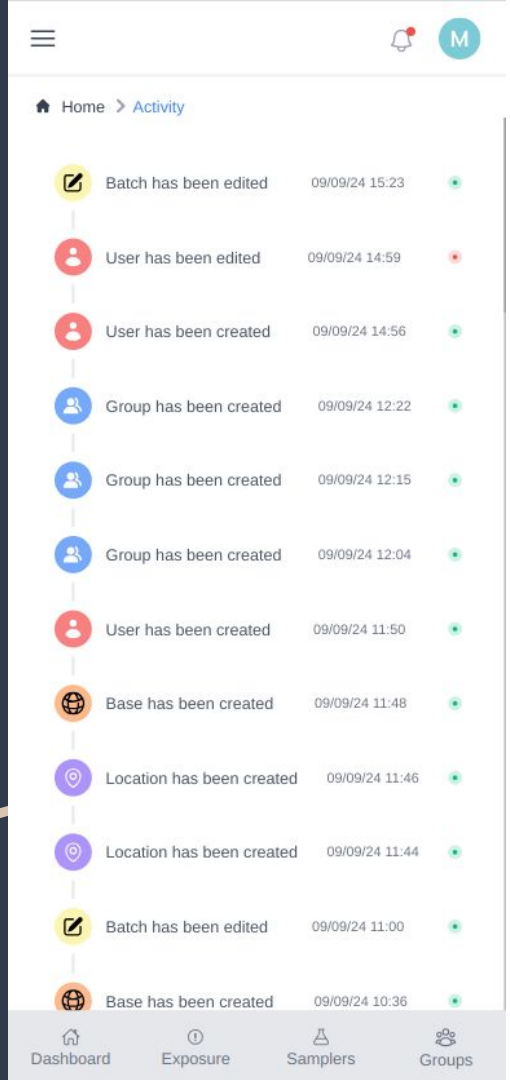
1. **Description:**
   a. **Add Location & Add Base Overview:** Provides users with simple forms to add new locations and bases to the system. These actions help in organizing and managing data efficiently.
2. **Key Points:**
   a. **Add Location:**
      i. **Enter Location Details:** Users can input details such as name, address, and other relevant information.
      ii. **Save & Sync:** Data is immediately saved and updated on the server if online, or stored locally if offline.
   b. **Add Base:**
      i. **Enter Base Details:** Users input information such as base name and location etc.
      ii. **Save & Sync:** Data is updated directly on the server when online, or stored locally if offline.
   c. **Offline Handling:** Both features store data locally in SQLite when offline.
   d. **Synchronization:** Automatically updates the server with new locations and bases once the user reconnects online.
3. **Flowchart or Simple Diagram:**
   a. **User → Open Add Location/Add Base Form**
   b. **Enter Details → Check Connectivity**
      i. **Online:**
         1. Directly Save and Update on Server
      ii. **Offline:**
         1. Store Data in Local SQLite
   c. **Reconnect Online:**
      i. Sync New Data to Server

Bulk-Turn



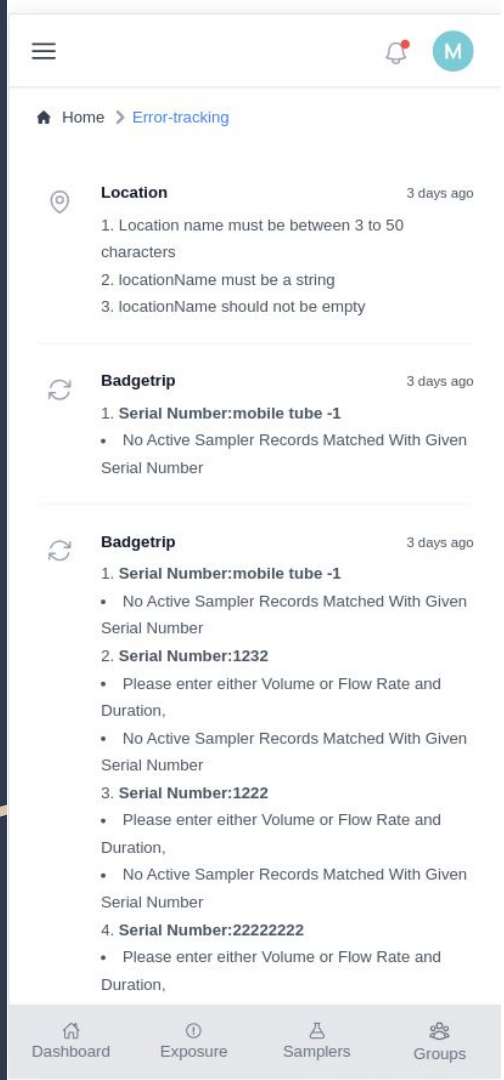
1. **Description:**
   a. **Bulk Turn In Overview:** Facilitates bulk processing by allowing users to download and upload sampler Excel sheets. Handles both online and offline scenarios for efficient data management.
2. **Key Points:**
   a. **Download Excel Sheet:**
      i. **Online & Offline:** Users can download a sampler Excel sheet to their device.
   b. **Upload File:**
      i. **Upload Process:** Users upload the Excel file to process bulk turn-in operations.
   c. **Online Handling:**
      i. **Direct Turn In:** When online, the system directly processes and turns in the samplers based on the uploaded file.
   d. **Offline Handling:**
      i. **Store Locally:** If offline, the uploaded Excel file is stored in local SQLite.
   e. **Synchronization:**
      i. **Reconnect Online:** Upon reconnecting, the system processes the bulk turn-in based on the locally stored file and updates the server.
3. **Flowchart or Simple Diagram:**
   a. **User → Download Sampler Excel Sheet**
   b. **Upload Excel File → Check Connectivity**
      i. **Online:**
         1. Directly Process and Turn In Samplers
      ii. **Offline:**
         1. Store Excel File in Local SQLite
4. **Reconnect Online:**
   a. Process and Update Bulk Turn In on Server

Historical Data Tracking



1. **Description:**
   a. **Historical Data Tracking Overview:** Allows users to track all actions performed within the app, regardless of whether they were done online or offline. This feature ensures transparency and data consistency by syncing actions when the user is back online.
2. **Key Points:**
   a. **Action Logging:**
      i. **Online Actions:** Automatically logs user activities (e.g., adding locations, turning in samplers, updating profiles) directly to the server in real-time.
      ii. **Offline Actions:** Actions taken while offline are stored in local SQLite for later synchronization.
   b. **Offline Handling:**
      i. Stores all actions such as adding new data, editing existing records, or uploading files locally when offline.
   c. **Synchronization:**
      i. When the user reconnects to the internet, all stored offline actions are synced to the server, updating the historical record with timestamps and details.
   d. **Complete Record:**
      i. Provides users with a full view of all actions performed, whether online or offline, with information on when data was synced.

Error Tracking



1. **Description:**
   a. **Error Tracking Overview:** Enables users to monitor any errors that occur when syncing offline data to the server. This feature helps ensure data integrity by alerting users of issues that need correction during the synchronization process.
2. **Key Points:**
   a. **Offline Data Creation:**
      i. Users can create or update data (e.g., adding locations, uploading files) while offline, and this data is stored in local SQLite.
   b. **Synchronization:**
      i. When the user reconnects to the internet, all locally stored data is uploaded to the server.
   c. **Error Handling:**
      i. If any issues occur during this sync (e.g., conflicts, missing fields, server rejections), the errors are logged and displayed to the user.
   d. **User Action:**
      i. Users can view the error details, understand the issue, and take appropriate action to correct it before re-attempting the sync.
3. **Flowchart or Simple Diagram:**
   a. **User → Create Data Offline → Data Stored in SQLite**
   b. **Reconnect Online → Start Sync**
      i. **If Sync Successful:** Data Uploaded to Server
      ii. **If Sync Error:**
         1. Log Error → Display Error to User

Data Synchronization

1. **Description:**
    a. **Synchronization Overview:** Describes the process where data created offline is uploaded to the server when a user logs in online, followed by downloading unsynced data from the server to the local SQLite database.
2. **Key Points:**
    a. **Offline Data Upload:**
        i. When the user logs in online, all data created or modified while offline (e.g., new users, samplers, or updates) is automatically uploaded to the server.
        ii. **Sync Trigger:** Upload process begins as soon as the user connects to the internet after working offline.
    b. **Server Sync:**
        i. After completing the upload, the system checks for any unsynced data on the server.
        ii. **Download Process:** The app fetches data from the server that hasn't been synced to the local SQLite database yet.
    c. **Local Update:**
        i. The downloaded data is set to the local SQLite database, ensuring that both server and local storage are fully synchronized.
3. **Flowchart or Simple Diagram:**
    a. **User → Logs in Online**
    b. **Offline Data → Upload to Server**
        i. Upload all data created or updated offline to the server.
    c. **Server Sync → Download Unsynced Data from Server**
        i. Download any data not yet synced from the server to local storage.
    d. **Set Data to Local SQLite → Sync Complete**

Generalized Workflow for Create/Update Operations

1. **Introduction**:
    a. The following workflow applies to the following operations:
        i. **Create/Update Group**
        ii. **Create/Update User**
        iii. **Create Location**
        iv. **Create Base**
        v. **Sampler Turn In**
    b. The workflow supports both **online** and **offline** scenarios, ensuring seamless operations regardless of connectivity.
2. **Key Workflow Sections**:
    1. Online Mode:
    a. **Direct Record Creation/Update**:
        i. Data is immediately **created/updated** on the **server** via API calls.
        ii. No local storage is used in this mode.
    b. **Dropdown Data Fetching**:
        i. Dropdown data (e.g., active clients) is fetched **directly from the server** in real-time.
3. **2. Offline Mode**:
    a. **Local Record Creation/Update**:
        i. The record is saved in **SQLite** with a flag isSynced: false.
        ii. Changes are stored locally and synced with the server when the user is back online.
    b. **Dropdown Data Retrieval**:
        i. Dropdown data is retrieved from **previously synced records** stored in the local **SQLite** database.