ChatGPT

Farmforce System UI/UX Analysis and Documentation

Overview of the Current UI/UX Structure

Farmforce’s platform consists of a **web application** for management and oversight and a companion **mobile application** for field data collection [1]. The web app is a cloud-based portal used by administrators, managers, and office staff to organize and monitor farming activities, while the mobile app (Android-based) is used by field staff to enter data in the field (even **offline**) and sync with the server later [1].

Web Interface Layout: The web UI follows a standard dashboard layout with a vertical navigation menu and a content area. A left-hand sidebar (or menu) lists the main modules – for example, *Origin Tenants*, *Producers*, *Fields*, *Trainings*, *Surveys*, *Case Management*, *Mobile Devices*, *Users*, etc. 【32†frame01】. Users navigate by clicking these sections, which loads the corresponding page in the main content pane. The header likely displays the Farmforce logo/environment name and user account options, while the main content area shows either data dashboards, lists, or form interfaces depending on the module. Overall, the web UI emphasizes data tables and forms in a utilitarian style, reflecting its focus on record-keeping and compliance data.

Mobile Interface Layout: The mobile app has a simplified interface optimized for on-the-go data capture. The home screen presents key functions for field agents – for instance, viewing a list of farmers (producers), capturing field information, filling out surveys, recording training attendance, etc. Navigation is typically via a menu (either a hamburger menu or a bottom navigation bar) that lists modules similar to the web app (Farmers, Fields, Surveys, etc.), allowing the user to jump to each feature. The mobile UI prioritizes offline-first data entry: screens are form-based with clear input fields, buttons to save/submit, and indicators for sync status (since the app runs offline and uploads data when connectivity is available [1]). The design on mobile is task-focused, with large tap targets and simple layouts (e.g. lists of records and detail pages) to accommodate use in rural field conditions.

Both web and mobile UIs are designed to be **user-friendly** and domain-focused, aiming to simplify digital data collection for agriculture programs [2]. However, the UI visuals are primarily functional; they favor clarity and completeness of data over sleek aesthetics. Below, we detail key workflows on each platform, identify usability issues, evaluate the design against usability heuristics, and recommend improvements.

Key Workflows and Navigation Flows

Web Application Workflows

- **Dashboard and Navigation:** Upon logging in, users are typically presented with a dashboard or home screen summarizing key metrics (e.g. number of farmers, active fields, pending compliance issues, etc.). This dashboard gives a quick overview of the system's status and may include charts or statistics. From here, the user can navigate using the sidebar. For example, clicking **"Producers"** in the sidebar opens the farmer management module.

- **Farmer Management (Producers Module):** In the **Producers** section, the interface displays a paginated list of farmers with identifying details. Each row in the list might show a farmer ID/code, name, location, and contact info【32†frame02】. Users can search, filter, or sort this list (for instance, filtering by region or cooperative). To view more details, the user clicks on a specific farmer's entry. This opens a farmer **profile dashboard** (or detail page) where comprehensive information is shown – personal details, farm locations on a map, linked family members, etc.【32†frame08】. The profile page may use tabs or sections to organize information (e.g. "General Info", "Fields", "Activities", "Surveys"). A small map widget might display the farmer's fields as GPS polygons overlayed on Google Maps [3]. From this page, managers can perform actions like editing farmer info or viewing historical data. Navigation breadcrumbs or a back button allow returning to the list.

- **Field Mapping and Details (Fields Module):** The **Fields** section allows users to see all field plots registered. Selecting **"Fields"** from the menu might bring up a map view or a list of field entries. A common pattern is a map on the top or side with pins or polygons for each field, and a list or filter to the side. Users can click on a field to see its details – e.g. field name, size, GPS coordinates, farmer owner – and verify it's within allowed areas. Given Farmforce's emphasis on geospatial data (integrating with Global Forest Watch for deforestation checks), the UI likely highlights whether a field lies in a protected zone [3]. Key workflow: an office user can upload or view field boundaries and ensure compliance (e.g. no fields in restricted areas). The layout for this might consist of a map panel and a details panel.

- **Training and Activity Logging (Trainings Module):** In **Trainings**, users track farmer training sessions. Navigating to "Trainings" shows a list of training events (with details like date, topic, location). Users can open an event to see which farmers attended and other notes. The workflow here would be adding a new training: clicking an **"Add Training"** button (often a plus icon) opens a form where details of a training session (date, trainer, topic, participants) are entered. This is saved and then appears in the list for future reference. Navigation allows switching back to the farmer's profile to see their training history, etc.

- **Surveys and Compliance (Surveys Module):** The **Surveys** module is crucial for compliance and certification data collection. On the web, this module might have two parts: one for **designing surveys** (for office staff) and one for **viewing survey responses**. The design interface lets authorized users create or edit survey forms (e.g. an Internal Inspection checklist or a Child Labor Monitoring questionnaire) – likely via a form builder UI. The response interface shows records of completed surveys from the field. For example, clicking **"Survey Responses"** might bring up a table of submitted surveys with columns such as survey title, farmer or field, date, status, and score/result【32†frame20】. The user can filter by survey type or search for a particular farmer's responses. Opening a survey response displays all answers in detail. In the demo, a child labor survey form is shown with many questions and responses listed sequentially【32†frame14】. The web UI presents each question (and answer) perhaps in a long scrollable form or a read-only view with labels like "Q1.4" and the question text【32†frame14】. This allows managers and auditors to review what field agents recorded. Navigation flows here include: designing a survey (creating new questions, setting scoring and branching), publishing it to mobile, and later reviewing the submitted results. Users can navigate between different surveys via dropdown or menu and export data if needed.

- **Case Management:** The **Case Management** module helps track issues or incidents (e.g. a child labor case or a deforestation alert). Navigating to this section shows a list of cases with status (open, resolved) and details. A user can click a case to view its information (related farmer, issue description, follow-up actions). The typical workflow: when a compliance survey or field report flags a serious issue, a case can be created. Through the UI, the manager would hit “New Case”, fill in details linking it to the farmer or survey record, and then use this interface to update progress on resolving the issue. The case management UI likely has forms for adding notes, assigning responsibility, and marking resolution. Users can switch between open and closed cases using filters.

- **Administration (Users, Devices, Tenants):** Admin users have additional menu options like **Users** (manage user accounts and permissions), **Mobile Devices** (register or monitor devices used by field staff), and **Origin Tenants** (manage organizational units or instances in a multi-tenant setup). The **Users** page shows a list of system users and allows creating or editing users (setting roles, reset passwords, etc.). **Mobile Devices** might list phones/tablets registered, showing last sync time, app version, etc., which helps in tech support. **Origin Tenants** likely appears only in a global (Orbit) context – listing different project instances or client organizations within Farmforce. In terms of navigation, these are typically only used by system admins, and each has its own straightforward form-based workflow (e.g. add a new user via an “Add User” form, or deactivate a device).

Navigation Flow Summary: The web app supports a **hub-and-spoke navigation**: users return to the sidebar to switch modules, and within a module they can drill down into details and then back out. Consistent header elements (like a home icon or breadcrumbs) allow quick return to the main dashboard. However, moving data between modules (for example, linking a survey result to a case) may require manual steps (copying identifiers or using filters) since the modules are separated in the UI. Overall, the web workflows revolve around viewing lists, clicking through to detail pages, and performing actions via forms or buttons at the top of those pages (e.g. “New”, “Edit”, “Export” icons). Familiar web conventions like pagination controls, search bars, and confirmation dialogs (when deleting records) are part of these flows.

Mobile Application Workflows

- **Logging In and Sync:** The field agent launches the Farmforce mobile app and logs in (if not already authenticated). On successful login, the app syncs any new data. The UI likely provides a **sync status indicator** (e.g. an icon or message showing last sync time) because offline capability is key. If the agent is online, new assignments or surveys from the server are downloaded at this point [1]. This ensures the mobile user starts with up-to-date data.

- **Home Screen and Navigation:** After syncing, the agent sees the home screen, which might be a simple menu or dashboard. This could be a list of tasks or modules: for example, buttons or list items for *Farmers*, *Surveys*, *Trainings*, *Purchases*, etc. In some versions, it might show a count of pending items (e.g. “Surveys to Complete: 2”). Navigation is likely via tapping one of these options. If a hamburger menu is used, tapping it would slide out a menu with the same module options. The design is minimalist for clarity in the field – large text labels and icons for each module for quick access.

- **Farmer Registration & Updates:** One core mobile workflow is registering a new farmer (producer) or updating farmer info during a field visit. The agent selects **“Farmers”** from the menu, which displays a list of farmers in their region (possibly searchable by name or farmer ID). To add a new farmer, the agent taps an **“Add Farmer”** button (often a plus **+** icon). The app then walks the agent through a multi-step form: entering personal details (name, ID, family info), taking a photo of the farmer, and capturing GPS coordinates for the farmer’s house or field. The UI uses standard mobile form controls (text fields, dropdowns for predefined lists like village or crop type, date pickers for birthdates, etc.). When capturing location or field boundary, the app likely opens a **map interface**: the agent can either drop a pin for a single coordinate or walk the perimeter of a field. The mobile UI would display a map with the current GPS position and allow the user to mark points to form a polygon for the field [3]. Throughout this workflow, the app provides prompts (e.g. “Stand at the corner of the field and press + to add a point”) and feedback on GPS accuracy. Once the form is filled, the agent taps **Save/Submit**, and the data is stored locally (with a queue to upload at next sync). The navigation then returns to the farmer list.

- **Field Data Capture (Activities):** Besides registration, field agents record ongoing **growing activities** (like planting, fertilizer application, harvesting) using the app. The agent might open a particular farmer’s profile on the mobile app – for example, by selecting the farmer from the list – which shows that farmer’s details and a menu of actions. From here, the agent can choose to add an **activity record**: e.g. log that “Farmer X applied pesticide Y on Date Z”. The UI likely presents an input form where the agent selects the field (if the farmer has multiple fields), chooses the activity type from a list (these correspond to things like planting, weeding, etc.), and enters required details (quantity, date, etc.). This workflow ensures data like chemical usage or farm practices are captured digitally on the spot. The app’s design for this uses simple lists and forms so that even users with basic smartphone experience can follow along.

- **Conducting Surveys/Inspections:** Another critical mobile workflow is completing **compliance surveys** (such as an internal audit, sustainability assessment, or a child labor monitoring survey). The field agent chooses **“Surveys”** (or a similarly named module like “Inspections” or “Assessments”) on the app. This might show a list of available survey forms or assignments. For instance, the agent could see “Farm Sustainability Assessment” or “Child Labor Follow-up Survey” in the list. Upon selecting a survey, the app may prompt to choose the target (e.g. which farmer or which field the survey is for) if not already specified. Then the app presents the survey questionnaire one question at a time or in pages. The UI for surveys on mobile focuses on ease of input: questions and answer controls are rendered clearly — yes/no toggles, multiple-choice options, number sliders, text fields for comments, etc. The agent progresses through questions by tapping “Next” after each, or scrolling if it’s one long form. For long surveys (some can have 100+ questions [4]), the app might break them into sections to avoid overwhelming the user. For example, sections with headings like “Child Details”, “Work Details”, etc., each with a subset of questions. The mobile app might display a progress bar (e.g. “Page 2 of 5”) to let the user know how much is left. During the demo, for instance, if the agent was filling the *Child Labour* follow-up survey, they would answer questions about the child’s awareness of activities, schooling status, hours of work, etc., just as listed in the form 【32†frame14】. The final step often requires capturing a **signature** or photo as proof; the app likely provides a signature pad for the farmer or agent to sign on-screen 【32†frame14】. Once all responses are entered, the agent taps **Submit**, and the survey is saved (with an indicator if it’s complete and waiting to sync). The app may validate that required questions are answered and show a confirmation like “Survey saved locally” or “Submission complete”.

- **Purchasing & Harvest Collection:** (If the app includes modules for input distribution or crop purchasing.) The agent might also record transactions such as delivering farm inputs (seeds, fertilizer) or collecting harvest from farmers. Workflows here would involve selecting the farmer, then entering transaction details (e.g. weight of crop collected, price, etc.) and possibly scanning a **barcode** on a bag for traceability [5]. The UI would be similar: a straightforward form with fields for quantities and a scan button to activate the camera for barcode scanning (if applicable). After submission, a receipt might be generated (the mention of *Printed Verifiable Receipts* suggests integration with portable printers [5], though that's more hardware than UI; still, the app might show a "Print" option if a printer is paired).

- **Syncing Data:** Throughout the day, the mobile app stores all new entries. When connectivity is available (mobile network or Wi-Fi), the agent triggers a **sync** (or it auto-syncs periodically). The UI should clearly indicate syncing – e.g. a status bar or sync icon animation – and notify the user of success or any errors. For example, after completing surveys or registrations, the agent might tap a **Sync** button on the home screen which then uploads all pending data and downloads any updates (like new survey assignments or farmer list changes). Feedback like "5 records uploaded successfully" or a warning for any failed records is an important part of this workflow, ensuring field users trust that their data was received by the system.

- **Viewing Information on Mobile:** The app also allows agents to view farmer information and history while in the field. For instance, before visiting a farmer, the agent can open that farmer's profile on the app to review past notes or verify their field locations. The profile screen on mobile likely shows key details (farmer name, ID, maybe a small map or list of fields, and recent activities or survey status). Navigation within a profile might use simple tabs or an expandable accordion (e.g. tap to see "Fields" or "Trainings"). This read-only access helps the agent have context during visits (like knowing if this farmer had any compliance issues before).

Overall, the mobile navigation is optimized for **linear task flows** – the agent typically completes one task at a time (e.g. fill a form, then submit) rather than multitasking. The interface uses large buttons, minimal text, and offline-friendly design (storing entries until sync). There is likely some redundancy built-in for reliability, e.g. confirmation dialogs ("Are you sure you want to submit?") and the ability to edit an entry before syncing if a mistake is noticed. The mobile and web workflows are complementary: data captured via these mobile flows (new farmers, survey results, etc.) becomes immediately available on the web platform for managers to review once synced.

UX Bottlenecks, Usability Issues, and Visual Inconsistencies

Despite covering the needed functionality, the current UI/UX has several pain points and inconsistencies that could affect user efficiency and satisfaction:

- **Complex Navigation and Cluttered Menus:** The web application's sidebar lists many modules (some with technical names like **"Origin Tenants"**) which can be overwhelming or confusing to users 【32†frame01】. Not all users will understand terms like *tenant* or need access to every module. The lack of clear grouping or hierarchy in the menu (e.g. mixing high-level concepts like tenants with daily tasks like surveys) is a usability issue. Users might have to hunt through a long list to find what they need, leading to navigation inefficiency.

- **Inconsistent Terminology (Farmers vs. Producers):** The system sometimes refers to farmers as "Producers" in the interface【32†frame01】, but other materials and user mental models use "Farmers." This mismatch can cause confusion. Similarly, internal codes or IDs (like *RA,PUR 038* in farmer IDs) are exposed in the UI【32†frame02】 without explanation, requiring users to recall or decode what they mean (e.g. RA,PUR might denote region and project). Using domain-friendly labels (e.g. "Farmers") and hiding or explaining codes would better match the real-world terminology.

- **Data Density and Readability Issues:** Many screens, especially on the web, display dense tables of data with minimal visual relief. For example, the farmer list shows multiple numeric identifiers, phone numbers, etc., in a grid fashion【32†frame02】. Important information (like the farmer's name) isn't distinctly highlighted, making scanning difficult. The lack of whitespace, grouping, or visual cues (such as icons or color coding) means users have to mentally parse large volumes of text. This can lead to oversight and increased cognitive load, especially when managing hundreds of records.

- **Visual Inconsistencies Across Modules:** Different modules exhibit slightly different interface styles or conventions. For instance, the **Survey response view** shows question codes like "Q1.5" alongside questions【32†frame14】, which is a very text-heavy layout, whereas other parts of the system (like basic farmer data forms) might not show any codes. Some pages might use different font sizes or input styles (possibly due to being developed at different times), leading to a lack of a unified look and feel. There might also be inconsistency between web and mobile in terms of iconography and workflows (e.g. the web uses a "+" icon for add, but mobile uses a text button, or vice versa). These inconsistencies can momentarily confuse users switching contexts.

- **Survey Interface Usability Problems:** The compliance survey forms are long and complex – the demo showed a form with dozens of questions and sub-sections【32†frame14】. On the web, presenting all questions and answers in one long scroll makes it hard to review or find specific information. Important cues (like which questions are unanswered or which ones failed compliance) could be lost in the wall of text. If field staff fill surveys on mobile in a similar long format, it can be tedious and error-prone. There's also evidence of **technical labels** (like conditional logic references "C1.1 = Q1.11" shown in the UI) being visible【32†frame14】 – something that should ideally be hidden from end-users. This suggests the survey module might not have a polished UI for either data entry or review, causing confusion and potential mistakes (e.g. agents might not understand branching questions if not clearly indicated).

- **Lack of User Feedback and Status Visibility:** It's not apparent if the system provides adequate feedback for certain actions. For example, when a user clicks to load a large list or save a form, there might be little indication of progress. In an enterprise system like this, operations (like importing data or syncing mobile entries) can take time; if the UI doesn't show a loading spinner or confirmation message, users could be unsure if the action completed. Similarly, on mobile, if an agent submits a form offline, they need clear confirmation that it's "saved locally" plus a reminder to sync – any ambiguity here is a UX risk. If these feedback mechanisms are insufficient, it violates the principle of keeping users informed of system status.

- **Error Handling and Validation:** Current UI screens likely have basic validation (e.g. required fields), but error messages might be technical or not very visible. For instance, if a user tries to submit a survey with a missing answer, does the interface clearly highlight the exact question to fix, or just show a generic error? Any ambiguity forces the user to search for the mistake, frustrating them. Also, if an error occurs during sync (say a network drop), the mobile app needs to inform the agent in plain language and queue the data – if it simply fails silently or shows a cryptic error code, field users might not know how to proceed.

- **Potential Performance Bottlenecks:** While not directly visible in static screenshots, enterprise users often face performance issues with large data sets. If the farmer list only shows 10 at a time, a manager might have to page through dozens of screens to find a record, which is inefficient 【32†frame02】. Also, maps with many field polygons could load slowly or become cluttered. If the UI does not virtualize or paginate effectively, users might experience lag, which is a UX issue (e.g. a map taking too long to render all points without a loading indication).

- **Mobile App Constraints:** On the mobile side, some usability issues include small touch targets or information overload on a small screen. If the mobile app tries to mimic the web's data density, it could result in very tiny text or controls. The presence of the phone's status bar in screenshots (time, battery) indicates the app doesn't use full-screen mode, which is fine, but also that agents see system UI elements that could distract. Also, if the mobile app relies on long lists (e.g. selecting a village from a list of 200 options), without search or proper filtering, that becomes cumbersome in the field. Finally, actions like capturing GPS points can be tricky – if the UI doesn't guide the user with accessible language and visuals (like "You are here" blue dot on the map), the agent might be unsure if they did it correctly. Any such hiccups slow down fieldwork and may require re-training.

In summary, the Farmforce system's UI is powerful but **could be more intuitive**. Users might struggle with navigation due to an overabundance of menu items, face cognitive load from dense data presentation, and encounter minor inconsistencies that erode confidence (for example, seeing an unfamiliar term or interface element and wondering if they're in a different system). These issues provide clear opportunities for refinement, as discussed in the recommendations.

Evaluation Using Nielsen's Usability Heuristics

Evaluating Farmforce's UI/UX against Nielsen's 10 usability heuristics highlights areas of strength and weakness:

1. **Visibility of System Status:** The system partially meets this heuristic. Positive: data changes (like adding a farmer or submitting a form) do result in updates visible in the list views immediately, so the outcome is shown. However, there is room to improve *feedback during processes*. For instance, when syncing data on mobile or loading a large dataset on web, users would benefit from clear indicators (e.g. a progress spinner or status message). Currently, feedback cues are not obvious, potentially leaving users wondering if an action was registered. Strengthening status visibility (loading animations, "last synced at…" messages [1], success confirmations) would improve trust in the system.

2. **Match Between System and the Real World:** Some parts of the UI align well with the agricultural context (e.g. using terms like *Fields*, *Harvesting*, *Farmers* in descriptions). But there are notable mismatches in terminology. The use of technical terms like *CMS_TENANT_ID* in filters or internal codes in the interface is confusing for users who expect real-world names 【32†frame02】. "Producers" vs

"Farmers" is another inconsistency – ideally, the UI should consistently use the term that end-users use in their daily language (likely "Farmers"). To comply with this heuristic, the system should replace or supplement codes with clear labels (e.g. show region names instead of codes, use "Organization" instead of "Tenant" if appropriate) and organize information in familiar categories (e.g. group data by growing season, since that's how farming activities are thought of, rather than by database technical groupings).

3. **User Control and Freedom:** Farmforce provides basic navigation freedom – users can switch modules freely via the sidebar and presumably log out or back easily. The mobile app likely allows field agents to save drafts of forms (giving control to complete a survey later). However, certain actions may lack easy "undo" or exit options. For example, if a manager accidentally initiates a new survey or opens the wrong farmer record, do they have a clear way to cancel or navigate back? The web UI could improve by offering "Cancel" buttons on forms (not just Save) and confirmation prompts on destructive actions (like deleting a farmer or case). On mobile, ensuring the agent can back out of a form without losing data (maybe autosave responses as they go) is crucial. Currently, the design doesn't obviously show undo features, so adding those would enhance user control.

4. **Consistency and Standards:** Consistency is moderate. The platform generally follows web and Android UI standards (e.g. forms with familiar controls, menu icons, etc.) and internal consistency could be better. There are slight variations in design between modules – for instance, the way an "edit" function is represented might differ in the Farmers module (maybe a pencil icon) versus in Surveys (perhaps an "Edit" button). Standards like using the trash bin icon for delete or a floppy disk icon for save should be applied uniformly. One inconsistency noted is the presence of system-coded labels in one module but not others, which breaks the consistency of a clean, user-friendly language. Additionally, aligning the mobile app's look and feel with the web (colors, icons, terminology) is important for a seamless experience. At the moment, a user switching from web to mobile might feel they are in two slightly different systems – a unified style guide and design system would address this.

5. **Error Prevention:** The system tries to prevent errors by structuring inputs (e.g. dropdowns for finite choices, which prevent invalid entries). For example, when adding a new farmer, using predefined options for village or crop type helps avoid typos or wrong data. Nonetheless, some improvements are needed. One scenario is multi-step processes like mapping a field: the UI should prevent errors by guiding the user (e.g. disabling the "Save" button until at least three GPS points are recorded for a polygon). Another scenario is data duplication – if an agent tries to register a farmer that already exists (maybe by national ID), the system should warn them. It's not clear if such checks are in place. Also, on web, when performing bulk operations like importing data, the UI should prevent mistakes (for instance, confirm "Are you sure you want to import? This might overwrite existing records."). Currently, these preventive measures are not visible in the demo, so reinforcing them would reduce user errors.

6. **Recognition rather than Recall:** The UI could do more to minimize the reliance on memory. At present, some features require users to remember context or codes. For example, if a manager is looking at a list of survey responses, each entry might just show a farmer ID and they have to recall which ID belongs to which farmer, unless they click through. It would be better if names were directly shown to aid recognition. The interface should display descriptive information wherever possible so users don't have to recall internal codes. On a positive note, the use of icons (if

consistent) can help quick recognition – e.g. a map pin icon next to a GPS field entry indicates location, which is immediately understood. To improve, the system should add tooltips or labels to ambiguous icons (for instance, an icon that looks like a funnel – is that “Filter”? A quick hover tooltip saying “Filter” would clarify). Designing screens such that options and results are self-evident (e.g. showing “(inactive)” next to inactive users in the list rather than requiring the user to recall color coding or another cue) will better support recognition.

7. **Flexibility and Efficiency of Use:** Farmforce serves a variety of users – from field agents with mobile phones to office analysts – so it should cater to both novice and expert use. Currently, it appears to offer one way of doing things for all users, without obvious shortcuts for experienced users. For example, an expert user might want to quickly navigate to a specific farmer or jump to a particular module via a keyboard shortcut or search. If the web app doesn’t have a global search or quick jump feature, that’s a missed opportunity for efficiency. Adding features like a type-ahead search bar that can find any record (farmer, field, etc.) from the top of the screen would let power users bypass menu navigation. Similarly, bulk operations (like selecting multiple farmers and marking them as having attended a training) would significantly speed up work for advanced users. The mobile app could allow experienced agents to clone a previous record or use templates (e.g. if they have to input the same training attendance for multiple groups). At present, the system seems to favor a step-by-step approach good for novices, but introducing optional advanced features (keyboard shortcuts, bulk actions, template entries, etc.) would improve efficiency for veteran users.

8. **Aesthetic and Minimalist Design:** The current design leans towards functionality over form. Screens are information-rich but border on cluttered, violating the minimalist principle. For instance, the survey detail page is essentially a text dump of all questions and answers【32†frame14】 – while thorough, it is visually overwhelming. A more minimalist approach would show only key information by default and hide or collapse the rest. The aesthetic could be modernized – perhaps the UI uses outdated visual styles (default HTML tables or basic bootstrap-like styling). There is little evidence of visual hierarchy (everything in the same font size, weight). This means important data doesn’t stand out and the interface can cause fatigue. Embracing whitespace, grouping related elements, and using typography (headings vs body text) would create a cleaner look. The mobile app likely has a simpler aesthetic due to platform guidelines, but the web UI especially could be streamlined. In short, **simplifying the visuals and focusing each screen on its primary purpose** (e.g. for a farmer list screen, the purpose is to find a farmer quickly, so the UI should prioritize search and key identifiers, with less peripheral info) would greatly enhance usability.

9. **Help Users Recognize, Diagnose, and Recover from Errors:** In an enterprise scenario, errors will happen (e.g. syncing fails, or an input is invalid). The system should gracefully handle these. It’s not clear how Farmforce currently surfaces errors. A good practice is to present an error message in plain language (e.g. “Network connection lost. Data will be saved and retried automatically.”) and offer a solution or next step. If currently error messages are cryptic (say, “Error code 5001: Save failed”), users would struggle. Also, when an error is highlighted (such as a validation error on a form field), it should be clearly marked (red outline, message explaining what to do). For example, if a number is out of an acceptable range, the form could say “Value must be between X and Y” next to the field. Recovery could be aided by allowing editing after submission in certain cases – e.g. if a field agent submits wrong data, perhaps an office user can correct it later with an audit log. Ensuring that every error feedback is actionable (not just informative) will help users recover. If the current system lacks in-line help or pointers for errors, adding those is important.

10. **Help and Documentation:** Given the complexity of the platform, new users benefit from on-screen help. It's not evident that Farmforce has context-sensitive help (like little "?" icons or a help sidebar). Users likely have separate training or manuals, but integrating help into the UI would follow this heuristic. For instance, a first-time user might not know what "Mobile Devices" module is for – a small info tooltip or an onboarding walkthrough could explain it. Providing a searchable help within the app (or at least links to an online knowledge base) would reduce dependency on external documents. Also, labeling is a form of in-app help – buttons with clear text labels ("Add Farmer") are self-documenting, whereas an unlabeled icon is not. Improving the labeling and adding brief descriptions (like a one-line description at the top of each module page about what it's for) would orient users better. If no embedded help exists yet, implementing guided tours for complex tasks (e.g. a guided "how to create a survey" overlay) could dramatically shorten the learning curve.

In summary, the Farmforce system meets some basic usability criteria but falls short on others, especially in terms of simplicity, consistency, and user guidance. The heuristic evaluation underscores the need for a more user-centered design approach moving forward.

Recommendations for UI/UX Improvements

Based on the analysis above, here are major recommendations to improve Farmforce's UI/UX. These suggestions target both specific modules and general design, aiming to enhance usability, efficiency, and visual clarity across the web and mobile applications:

Field Data Capture (Mobile App)

- **Simplify Mobile Navigation:** Implement a clearer navigation structure in the app. For example, use a **bottom navigation bar** with 3-5 primary sections (Dashboard, Farmers, Activities, Surveys, Profile) for quick access, instead of hiding everything in a side menu. This way, field agents can switch between common tasks with one tap. Less-frequent options (settings, sync logs) can remain in a hamburger menu. The goal is to make the most used features immediately visible.

- **Guided Workflows and Offline Indicators:** Introduce guided task flows for common actions. For instance, a **"New Farmer" wizard** that breaks the registration into steps (Personal Info -> Field Mapping -> Photo -> Review) with a progress bar can make the process more approachable. Clearly indicate offline status and syncing: use icons or color changes (e.g. data waiting to sync could show an orange dot, which turns green once synced). A status banner like "Working offline – changes will sync when connection is available" reassures users that the app is functioning as intended [1].

- **Enhanced Data Input on Mobile:** Make data entry forms smarter to reduce errors and effort. Auto-fill known information where possible (e.g. if an agent selects a village for one farmer, remember it as a default for the next farmer in the same session). Use **mobile-friendly input controls**: large toggles for yes/no questions, star rating visuals for quality scores, date pickers for dates – this speeds up entry. For numeric inputs like weights or prices, allow using the phone's numeric keypad. Ensure labels are clear and concise, avoiding any technical jargon.

- **Improve Map & GPS UX:** Field mapping is a complex task on mobile, so improve this interaction. Use the phone's capabilities: for example, show the agent's current GPS accuracy and prompt them to wait if the signal is weak. When recording a field boundary, display each point on the map with a line connecting to the last, giving immediate visual feedback of the polygon shape. Allow an “Undo last point” action in case they tap incorrectly. If possible, integrate a satellite imagery toggle so agents can see field outlines from aerial view which helps in tracing boundaries. All map interactions should have on-screen instructions (e.g. “Tap + to add a waypoint” or “Double-tap to finish drawing”) so the user isn’t guessing.

- **Contextual Validation and Tips:** Provide real-time validation on mobile forms. For example, if a user enters an unusually large farm size, show a tip like “Is this acreage correct? The average in this area is much smaller.” This kind of gentle validation can catch data entry mistakes on the spot. Also, include small help icons next to complex fields – tapping a help icon could show a definition (e.g. what counts as a “Farmer Group ID”). This reduces the need for external reference to manuals.

- **Leverage Phone Hardware:** Incorporate features like barcode scanning and camera use more seamlessly. Since Farmforce supports bag barcode tracking [5], the app could have a **“Scan” button** prominently in any form where a code is required, which opens the camera quickly. Similarly, allow attaching photos in surveys or farmer profiles (e.g. photo of an ID card or a farm issue) – and make sure the UI shows thumbnails of taken photos for confirmation.

- **User Guidance and Training (In-App):** Include a brief tutorial mode or tooltips for new field agents. For example, on first login, highlight “This is the Sync button – tap here to send your data when you have network” with a little arrow. Provide a “refresh” or “sync now” button that’s clearly labeled, and maybe a log that shows last sync time and records synced, for transparency. This will build confidence in using the app correctly and avoiding data loss.

Dashboard and Data Visualization (Web App)

- **Customizable Dashboard Views:** Redesign the web dashboard as a **flexible, widget-based dashboard**. Users should be able to see key metrics at a glance and customize what’s shown based on their role. For instance, a project manager might see “Total Farmers Registered” and “Surveys Completed This Month”, while an agronomist might see “Fertilizer Applications This Week” or a map of fields. Implementing draggable, resizable widgets (charts, KPIs, recent activity lists) that users can add or remove will cater to diverse needs. At minimum, provide distinct sections for major areas: e.g. Farmers, Compliance, Operations each with a summary (like *X new farmers this month*, *Y open compliance cases*).

- **Improved Information Hierarchy:** Within dashboard widgets and list pages, highlight the most important data with visual weight. For example, use **large, bold text for critical numbers** (like number of active farmers) and smaller text for labels or supporting info. Use color-coding to draw attention: green for on-target metrics, red for areas needing attention, yellow for warnings. On list pages (e.g. farmer list), consider switching to a card or tile layout for a more scannable presentation – each farmer could be a card showing Name, ID, village, and an icon or badge if any issue is flagged (like a red exclamation if they have an open case). This card layout with spacing can be easier to read than a tight table.

- **Data Visualization for Trends:** Incorporate charts and graphs for key data trends directly in the UI. For example, on a *Harvests* or *Purchases* page, include a small line chart of total volume over time. On the compliance surveys overview, show a bar graph of compliance scores or percentage of farmers compliant vs non-compliant. Visualizing data makes it more digestible than raw numbers. Ensure these charts have clear legends and titles. Users should be able to click on a segment of a chart (say the slice of “non-compliant farmers”) to drill down to the underlying records – this tight integration between visual summary and detailed data helps users navigate efficiently.

- **Responsive and Accessible Design:** Update the web UI to be fully responsive so that it works on various screen sizes (in case managers open it on a tablet or small laptop). That means the sidebar might collapse into a hamburger menu on narrow screens, and cards or tables reflow. Use larger clickable areas for any interactive element (buttons, table rows) to improve accessibility. Also ensure proper contrast in the color scheme (currently, if any text is light gray on white, that’s hard to read; use dark text on light background for primary content). Add accessible labels and alt text so that screen reader software can describe dashboard elements, improving usability for users with disabilities.

- **Key Performance Indicator (KPI) Highlights:** Dedicate part of the dashboard to KPIs that update in real-time (or near real-time). For example: “Farmers Registered: 1,250” with a small up-arrow showing an increase from last week. Or “% Farms Mapped: 87%” giving a quick sense of progress. These should be visually prominent (perhaps as a horizontal band of metric cards at the top). This reduces the need for managers to run reports elsewhere – the most vital stats are front and center. It also gives immediate feedback on field activities (e.g. if field staff just uploaded new surveys, the compliance rate metric might change, showing the system is up-to-date).

- **Streamline Lists and Filtering:** For pages that still need to show lists or tables (which is many of them), enhance the filtering and batch operations. Provide a **multi-filter sidebar** where users can apply common filters easily (e.g. filter farmers by region, crop, or registration date with checkboxes or dropdowns, rather than having to enter filter queries). Show the active filters clearly and allow one-click clearing of filters. Additionally, add batch action buttons above tables – e.g. “Export List” (to CSV) or “Send SMS to Selected Farmers” if communications are integrated. This way, the dashboard not only informs but also becomes a launching point for action.

Mapping Tools and Geospatial Features

- **Integrated Map Dashboard:** Create a dedicated **mapping view** that can be accessed from the main navigation (or dashboard) to visualize geospatial data. This could be an interactive full-page map showing all farmer locations and field boundaries. Include layer controls so users can toggle what’s visible (e.g. fields, collection centers, deforestation risk layers). For global managers using Orbit, allow switching the view between different project “tenants” on the map easily. This single map view would give a holistic spatial overview, complementing the data tables.

- **Enhanced Field Detail View:** When viewing an individual field (on web or mobile), improve how information is presented. On web, instead of just textual coordinates or a link to a static map, embed an interactive map snippet that highlights that field polygon. Display key attributes next to the map – e.g. field size, crop type, last inspection date. Use subtle highlights: if a field is flagged (e.g. overlaps a protected area), outline it in red on the map and show a warning text like “This field is in a no-grow zone”. This direct visual cue is more effective than burying that info in text. Moreover, provide a quick action like “View in full map” to jump to the full map page centered on this field.

- **User-Friendly Mapping Interface (Web):** For office staff editing or adding geospatial data, create a user-friendly map editing tool. For example, if an office user needs to adjust a field boundary, they should be able to drag vertices on a map with the mouse, rather than inputting coordinates manually. Provide satellite imagery background as an option (since seeing actual field outlines can guide manual adjustments). Add a measuring tool so users can verify distances or areas on the map easily. All map tools should use standard icons (a hand icon for pan, a crosshair for adding a point, etc.) and have tooltips. This encourages staff to directly use the mapping tool without needing GIS expertise.

- **Performance and Clarity in Maps:** Large numbers of points or polygons can clutter the map. Implement **clustering** for farmer points (e.g. show a single marker with "+20" when zoomed out, which splits into individual markers as you zoom in). For polygons, perhaps only outline fields when zoomed in to a certain level, to avoid a map full of overlapping shapes. Use distinct colors for different data layers: for instance, farmer home locations could be blue markers, field polygons green, and if there's a layer for deforestation alerts, those could be red areas. A legend box on the map should explain these colors clearly. This way, the map remains interpretable even as more data is added.

- **Link Maps with Data Actions:** Make the maps interactive with the rest of the UI. For example, on the map dashboard, if a user clicks a farmer's marker, show a tooltip with their name and a link "Open Profile" which jumps to the farmer's profile page. Similarly, from a farmer's profile, a user could click on a field name and choose "Show on Map" to be taken to the global map with that field highlighted. This cross-linking ensures that spatial data isn't siloed – it becomes another intuitive way to navigate the system (some users think visually and would prefer to find a farmer by location rather than by name).

- **Mobile Map Optimization:** On the mobile app, ensure the map usage is optimized for small screens and intermittent connectivity. For example, allow the mobile app to cache map tiles for the areas of interest (so that agents can see the map even offline). Provide a **compass and "current location" button** on mobile maps so agents can orient the map to their direction, which is helpful when walking a field. If multiple fields are near each other, the mobile map could label them or allow tapping one to see the farmer's name – reducing confusion if agents work with neighbors. Simplifying the map UI on mobile (perhaps only showing necessary controls) will prevent accidental taps.

Compliance & Case Management Improvements

- **Streamlined Survey Design UI:** Redesign the **survey builder** (if one exists on web) to be more intuitive. Instead of a bare-bones form listing questions sequentially, use a drag-and-drop interface for building surveys. For example, office staff could drag question types (text, multiple-choice, number, photo, signature) onto a form canvas. They should be able to visually see the structure of the survey, group questions into sections, and set conditional logic using a clear rules interface (e.g. "If Q5 = No, then show Section X" – phrased in plain language). This would eliminate the need for showing raw question codes and make the creation of complex surveys (like a 100-question certification survey) more user-friendly [4]. Templates for common surveys could be provided to further ease the process.

- **Improved Survey Filling Experience:** Whether on web or mobile, the UI for **conducting surveys** should focus on one question at a time to avoid overwhelming the user. On the web app, if office staff or auditors ever input survey data directly, use a multi-step form or at least accordion sections to collapse groups of questions. On mobile (for field staff), consider a **page per section** approach: e.g. 10 questions per page with a “Next” button, rather than a single long scrolling list. Provide a progress bar or percentage to motivate completion. Also, implement validation as the agent goes (highlight unanswered required questions before allowing them to finish a page). If a survey is partially saved, mark clearly which sections are incomplete when resuming. These improvements will reduce errors in responses and fatigue.

- **Contextual Guidance in Surveys:** Given the complexity of some compliance questions, embed guidance into the survey UI. For example, near a question about hazardous work, include an info icon that when tapped, displays the definition of “hazardous work” so the agent knows what to look for. This way, even if the survey is lengthy, the field agent doesn’t need to remember training material – the app guides them. For numerical questions, show units (kg, hectares, etc.) clearly next to fields to avoid confusion. These small touches prevent misinterpretation of questions and ensure higher quality data.

- **Link Surveys to Cases Automatically:** Bridge the gap between surveys and case management. The system can be smart: if a survey response indicates a serious issue (e.g. child labor confirmed, or a score below a threshold), prompt the user (or automatically) to create a **case** for follow-up. For instance, after an auditor marks “Non-compliant” on critical questions, the UI could show: “This farmer is non-compliant. Create a new Case for remediation?” – with a one-click action that generates a case record pre-filled with relevant info (farmer ID, survey reference, issue type). This saves time and ensures no compliance issue falls through the cracks. The new case could then be accessible in the Case Management module for tracking.

- **Case Management Dashboard:** Revamp the case management section to function as a task dashboard for issues. Cases should be sortable and filterable by priority, type, region, etc. Use visual markers to indicate urgency – e.g. overdue cases in red, upcoming deadlines in amber. Each case entry in the list could display key info like farmer name, issue summary, and days open. Provide an option to view cases on the map as well (if location is relevant, like cases tied to fields). Within a case detail view, use a timeline layout: show an activity log (when it was created, notes added, steps taken) in chronological order. Include fields for assigning the case to a specific staff member and for target resolution date. This structured, visually clear approach will make it easier for managers to oversee multiple investigations or remediations at once.

- **Notifications and Alerts:** Add a notification system for compliance-related events. For example, if a new case is created or if a survey submission has a critical finding, relevant users (managers or compliance officers) should get an alert in the web app (and possibly via email). A bell icon on the web interface could show unread notifications – clicking it drops down a list like “Case #123 overdue for Farmer X” or “Survey indicates possible deforestation in Field Y”. On mobile, if field agents need to be aware of something (like “a re-inspection is required for Farmer X”), the app could have a Notifications section. This proactive alerting improves the responsiveness of the organization to data the system is collecting.

Visual Layout, Information Hierarchy, and Interaction Design Enhancements

- **Modernize the Look & Feel:** Update the visual design using a modern design system – for instance, adopt a clean, flat design with a consistent color palette that reflects the Farmforce brand (earthy greens for agriculture, etc.). Use **consistent typography**: define heading fonts and body fonts, and use them uniformly across modules. Currently, some pages may not differentiate heading text from normal text clearly, so establishing this hierarchy will make screens easier to scan. Simplify color usage – maybe one primary accent color for actions (e.g. Save buttons in a distinct color), one secondary color for info messages, and red for errors. Too many colors or inconsistent shades should be eliminated to avoid a patchwork look.

- **Consistent Iconography and Labels:** Create or choose a cohesive icon set for use throughout the application (both web and mobile). Every common action or entity should have a recognizable icon (e.g. a person icon for farmers, a leaf or tree for fields, a clipboard for surveys, a flag or exclamation for cases). Place icons alongside text labels in menus and important buttons to leverage both visual and text cues. Ensure that icons are used consistently – for example, if a pencil icon means "edit" in one module, the same icon should mean edit everywhere. Also, audit the language used in UI labels and tooltips to ensure consistency (if one page says "Add New" and another says "Create", choose one term and standardize it). This uniformity will reduce user confusion and learning curve, as the interface behaves predictably.

- **Improve Information Hierarchy on Pages:** Redesign pages to emphasize what's most important. For instance, on a farmer profile page, the farmer's name and status (active/inactive, or compliant/non-compliant) should be at the top in a prominent way, possibly next to a profile photo. Less critical details (like internal IDs or registration timestamps) can be smaller or lower on the page. Use grouping and section headers to break up information. In a long details form, insert subheadings (with a slightly shaded background) to denote "Contact Information", "Farm Details", "Certification Status", etc. This acts like a wireframe that visually separates content into digestible chunks. Users will find it much easier to locate specific info with this structured layout.

- **Use Cards and Panels for Clarity:** Where appropriate, replace plain tabular sections with **card-based layouts**. For example, instead of listing all family members of a farmer in a flat list, show each family member as a small card (with their photo, name, age, relation) – this makes the data more tangible and easier to browse. Panels or cards can also be used for upcoming trainings, recent activities, etc. Cards create natural spacing and are a familiar UI paradigm (used in many modern apps) that can highlight key data points and allow quick actions (like an "Edit" pencil on the card).

- **Interactive Elements and Feedback:** Enhance the interaction design by providing immediate visual feedback on user actions. For instance, when a user clicks a button, give a subtle animation or color change to confirm the click was registered. If a form is successfully submitted on web, perhaps momentarily highlight the new entry in the list (e.g. yellow flash that fades) to draw attention to it – this kind of micro-interaction confirms to the user that their action succeeded and shows where the data went. On mobile, incorporate haptic feedback for certain actions (a slight vibration on a long-press or when a GPS point is recorded) to provide a non-visual confirmation. These small interaction improvements make the system feel more responsive and alive to the user's inputs.

- **Simplify and Standardize Workflows:** Review multi-step workflows and remove any unnecessary steps or clicks. For example, if adding a training requires navigating through three different screens, see if it can be done in one pop-up dialog or a single page with sections. Standardize the workflow design: ideally, all “create new” actions open in a similar style – either a slide-out drawer or a modal dialog – rather than sometimes a new page and other times a pop-up. This consistency will mean once a user learns how to add one type of record, they can apply that knowledge elsewhere. It also reduces the sense of complexity. As a wireframe concept, imagine a unified “New Item” panel that changes form depending on context (new farmer, new field, new training, etc. but always with the same layout style).

- **Accessible Design Practices:** Implement best practices for accessibility which will also improve overall UX. Use sufficient contrast for text and interactive elements (e.g. lighten the background or darken text colors if needed). Ensure that form controls are labeled (both visually and in HTML for screen readers). Provide keyboard navigation support on the web app: users should be able to tab through fields in a logical order, and use arrow keys to navigate table rows or dropdown lists. This not only helps users with disabilities but also those who prefer keyboard usage. Also consider localization/internationalization – if Farmforce is used in multiple languages, design the UI to accommodate longer text strings gracefully and consider cultural differences in icons/colors.

- **Prototype and Solicit User Feedback:** Before fully implementing these redesigns, create low-fidelity wireframes and high-fidelity prototypes for key screens (dashboard, farmer profile, mobile home, survey form, etc.). Involve actual users (field agents, office staff, admins) in testing these prototypes. Their feedback will be invaluable – for instance, they might indicate that certain data should be more prominent or that a workflow is still too convoluted. Iteratively refine the design with this feedback. This user-centered design approach will ensure the end result addresses real user needs and pain points, leading to higher adoption and satisfaction.

By applying these improvements – simplifying navigation, clarifying layouts, ensuring consistency, and enhancing interactivity – Farmforce’s UI/UX will become more intuitive and efficient. The system will not only meet its functional requirements (securing first-mile data) but do so in a way that delights users, reduces training burden, and increases productivity. Ultimately, a well-designed UI/UX will empower both field staff and office users to leverage Farmforce’s full capabilities, driving better data quality and smoother operations across the agricultural value chain.

Sources:

- Farmforce website and materials for context on system usage [2] [1].
- Observations from the Farmforce system demo video (web & mobile UI flows, August 2025).
- Nielsen’s Usability Heuristics (1994) for theoretical framework.

---

[1] [5] Farmforce | PPTX
https://www.slideshare.net/slideshow/farmforce-36967531/36967531

[2] Farmforce - Tackling Food's First Mile
https://farmforce.com/

[3] [4] Farmforce Solutions: Farmforce Origin (IMS)
https://farmforce.com/products/information-management-system-ims/