Software Requirements Specification (SRS)

Document

Offline data collection and cleaning app

Team - 38

Aishita Marri

Harshikaa Agrawal

Kushal Balabhadruni

Sriyansh Suryadevara

Vishesh Saraswat

Brief problem statement

To develop a mobile application which will be used to fill survey forms in remote locations where internet connection is not available. The data can be synced when internet connection is back. The form should implement skip logic and data validation to reduce redundancy and the time spent on data cleaning. There should also be a website which is used for form creation by uploading form excels, data organisation and display.

System requirements

The mobile app will work with supported Android/iOS mobile phones.
Expo Go app required on phone for testing out how the app is functioning currently
Mobile app development done with react-native. Web Dashboard Development done with MERN.
Material design used to maintain consistency.

Users profile

1. Person filling the form on behalf of others
   - Should be able to fill the form multiple times
   - Is aware of some specifics about the app like presence of skip/conditional logic
   - Understands the usage of the technology and how MCQs work
   - May not know English
2. Data Analysts for data cleaning and analysis
   - Requires an intuitive way of visualising data

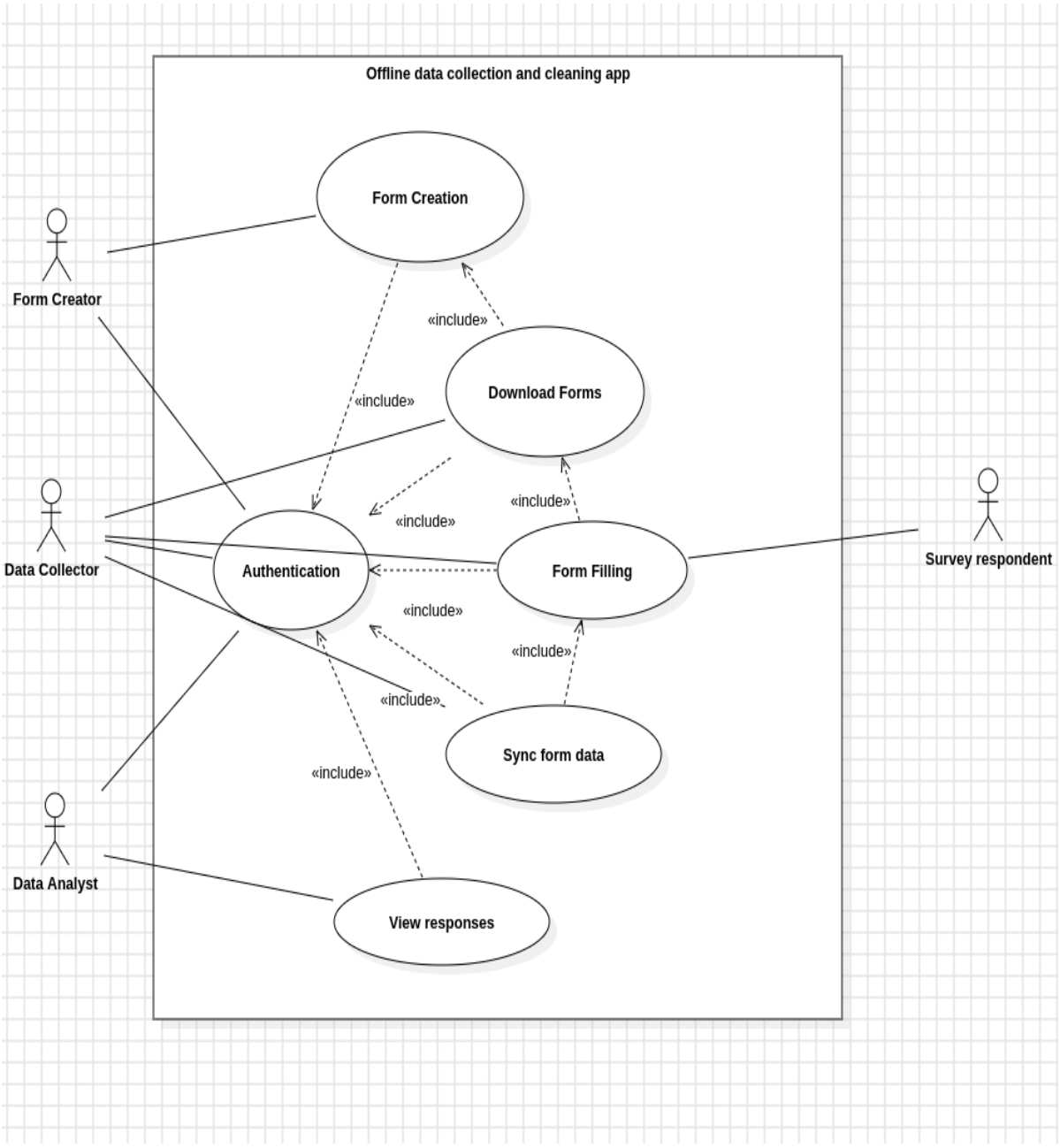
- Empty columns should not be there as specified by the client

3. Someone who is creating the form itself
   - Expects a simple interface (method) to add skip/conditional logic in the questions

Feature requirements (described using use cases)

No.	User Case Name	Description	Release
1.	Make Forms	Enable creation of forms from excel sheets.	R1
2.	Downloading Forms	View all forms the data collector has access to, download them when there is access to Internet.	R1
3.	Fill Forms	Interface for users to fill in forms downloaded.	R1
4.	Sync Form Entries	Sync entries filled offline with the server when network connection is present.	R1
5.	Authentication	Ensure users can only view/download forms that they have acess to.	R2
6.	View Responses	View responses on the form visually (data analytics) and export to csv/xlsx.	R1/R2

Use case diagram



Use case description

Use Case Number:	UC-01
Use Case Name:	Make Forms
Overview:	Users would need to have an option to create forms which will later be given to data collectors to collect data from the field.
Actors:	Form Creator
Pre condition:	1. User must have an existing account through which they can create forms 2. User must be aware of the xlsx file format used to create forms
Flow:	Main (success) Flow:- 1. User opens the web dashboard for creating forms 2. User can install the template using which they can create their form. 3. They can upload a .xlsx file in a format specified in the template. 4. The file is parsed and made into a form 5. The user then adds this form to a group 6. Data collectors can download and use this form
	Alternate Flows: 2a. The file is not formatted as specified 1. User is shown an error message with details on what is wrong 2. They can then upload the file again and start from 2. 5a. User wishes to delete form 1. They can remove the form from the group and stop accepting more responses
Post Condition:	Access can be given to data collectors who will be using this form and they can download it if required.

Use Case Number:	UC-02
Use Case Name:	Downloading Forms
Overview:	Field data collectors should be able to download the forms they have access to while connected to the internet. All downloaded forms should remain accessible offline, allowing collectors to start filling them out even without an internet connection.
Actors:	Field Data collectors, Form Creators
Pre condition:	1. The required form must be created and stored in the database following the specified syntax. 2. The data collector must be authenticated to access the form
Flow:	Main (success) Flow: 1. User logs in with his/her credentials. 2. All accessible forms are visible in the download forms page 3. User clicks on download form, which makes the form appear in a different list of forms available for filling offline
	Alternate Flows: 1a. User authentication fails - Error message displays the issue: credentials are incorrect, no internet connection or internal server error - User can retry after taking the necessary steps 3a. Download fails - Corresponding error message is flashed: network connectivity, local file creation-copying issue etc. - Partially downloaded forms are discarded from local storage and do not appear on the list of forms available to be filled. - User has to retry download
Post Condition:	Forms that the user has access to and has downloaded will be stored locally. These forms will appear in a separate tab within the app's interface, listing them as available for filling. The user can start filling out any of these forms at a later time.

Use Case Number:	UC-03
Use Case Name:	Fill forms
Overview:	Users should be able to fill out a form (which is available to them), save the data locally, and have the saved form displayed in the "Filled Forms"

	section. This allows for data collection in remote areas, with synchronization to the server when an internet connection is available.
Actors:	1. Field data collectors.
Pre condition:	- The user is authenticated and logged in. - The user has downloaded one or more forms that are available for filling. - The selected form is accessible and ready for input.
Flow:	Main Flow 1. User picks a form from the available forms. 2. The first question asks for consent, and if provided starts recording the conversation between the surveyor and the person being surveyed 3. The user starts filling the form. 4. Once completed, the user can save the form. 5. The form will be saved in the “filled forms” page and should be ready to sync when required.
Alternate Flow:	Alternate Flow 1a. The user selects a partially filled form from the “filled forms” section. 1. The app should display the entries which were previously filled and give option to edit them 2. Also user should be able fill new entries and go to 3. 2a. The user inputs an incorrect value (e.g., a character in a numeric field), the system detects the error 1. The app should display an error with appropriate error message. 2. User corrects the field and continues filling the form 3a. Saving the form midway. 1. The user opts to save the form before completing all fields. 2. The system saves the current state of the form locally. 3. The system also stops the audio recording and saves it locally. It resumes when the saved form is opened again to continue. 4. The partially filled form remains in the "Filled Forms" section. 4a. The audio recording fails: 1. The user is shown an error message 2. The response is stored without the audio
Post Condition:	The filled (or partially filled) form is successfully stored in local storage and is visible in the "Filled Forms" section. The form data is intact and ready for future editing or synchronization with the server when an internet connection is available.

Use Case Number:	UC-04
Use Case	Sync Forms

Name:	
Overview:	Data collectors can sync filled entries with the main server which is then stored in the database
Actors:	- Form Creator can view these saved entries - Data Collector can choose to upload these entries
Pre condition:	1. Filled form entries should be available on the mobile device 2. The device should be connected to the internet
Flow:	Main (success) Flow: 1. The data collector goes to the sync forms section 2. They review the filled entries 3. They press the sync button 4. Form entries are sent to the server and then deleted from the mobile device 5. The form creator has access to the entries now
	Alternate Flows: 2a. The data collector wants to delete a saved entry 1. They press on the delete button next to an entry 2. The entry gets deleted from their mobile device 3a. Mobile device is not connected to the internet 1. An error message is shown saying no network connectivity 2. The user can try again after they have internet access
Post Condition:	The form creator can now view the saved entries, export them to a spreadsheet and view data visualisations.

Use Case Number:	UC-05
Use Case Name:	Authentication
Overview:	Ensure data privacy and control access to forms.
Actors:	Admin, Form filling volunteers
Pre condition:	Account must exist
Flow:	Main (success Flow: 1. Login page is opened 2. Correct log in details are submitted as login input 3. App / Website opens
	Alternate Flows:

	1. a. Incorrect login details are submitted as login input b. Redirected to a page which says incorrect credentials 2. a. Any url except homepage is opened b. Not already Logged in c. Redirected to Log in page
Post Condition:	For App: Access to filling and downloading form For Website: Access to make forms, data analysis, downloading data

Use Case Number:	UC-06
Use Case Name:	View Responses
Overview:	Data Analysts and Form Creators can view the responses submitted by people and visualisation about the data
Actors:	Form Creator & Data Analysts
Pre condition:	1. There must be an existing form created 2. The form must have responses collected 3. The user must be logged in with correct credentials that have access to form data
Flow:	Main (success) Flow:- 1. The logged in user has forms they have access to the data of 2. They click on a form 3. They have an option to export and download the data 4. There is also an option to view visualisations
	Alternate Flows: 3a. The user wishes to filter out particular columns (to get only responses for questions that were actually answered): 1. The user has options to only download responses where user filled a particular response to some questions 2. They choose the options that they want 3. A filtered file is exported 4a. Text based questions: 1. In case of text based questions visualisations the user is informed if visualisations are not available
Post Condition:	The users can make inferences and analyse downloaded data according to their wishes