Technical Specification - NFRI Predict Web Application

Last revision: 30 Jun 2022

Authors:

Simon Bowen, Tom Feltwell
Open Lab, Newcastle University
{simon.bowen | tom.feltwell}@newcastle.ac.uk

Technical Review:

James Hodge
Open Lab, Newcastle University
j.hodge1@newcastle.ac.uk

Summary

This document describes the technical specification for the *Web Application* (web interface and supporting application) for the NFRI Predict tool, which was designed as part of the Collective Crisis Intelligence project between October 2021 and June 2022. The project was run by Nesta, in collaboration with the International Federation of Red Cross and Red Crescent societies (IFRC) and Open Lab, Newcastle University. The Web Application was designed for use by the Nepal Red Cross to assist their planning and distribution of non-food relief item (NFRI) packages such that their contents better reflected the needs and values of affected communities. This document describes the proposed Web Application design and a detailed proposed technical implementation of the Web Application and supporting system.

Intended Audience

This document describes the technical design and proposed implementation for the Web Interface and supporting web application, designed as part of the Collective Crisis Intelligence project. This document is intended for UX/UI designers, developers, and system architects who would implement the Web Application. This document details a proposed system architecture, breakdown of all technical components and an estimate of tasks and associated time and financial costs to implement the system.

Acronyms/Glossary

Term	**Definition**
API	Application Programming Interface
CCI	Collective Crisis Intelligence (Project)
NRCS	Nepal Red Cross Society
IFRC	International Federation of Red Cross and Red Crescent Societies
RC	The Red Cross
NFRI	Non food relief items - items other than food or money distributed after a crisis, e.g. blanket, bucket, soap, clothing.
ML	Machine Learning

System Requirements

From conversations with NRCS staff it was determined that an interface to NFRI Predict should be:

- Independent of rather than integrated with existing applications used by NRCS (Microsoft Power BI and PointNemo) to enable NCRS teams to evaluate where and how NFRI Predict outputs would be integrated with other data and visualisations;
- Web-based to better integrate with the web-based NFRI Predict model;
- Designed for the desktop PCs typically used by NRCS staff - using the Microsoft Windows 10 operating system and display resolution of 1366 x 786 pixels (16:9 HD)
- Designed for the three web browsers typically used by NRCS staff - Google Chrome, Mozilla Firefox, Microsoft Internet Explorer

System Overview

The NFRI Predict model predicts the likelihood (or probability) of individual NFRI items to be considered essential by a specific household (characterised by certain geographic, demographic, health-related, and cultural features) based on training data gathered from multiple households. The Web Application would enable NRCS staff to query and display results from the NFRI Predict tool during the preparedness and response phases of the emergency management process. A web interface to the NFRI Predict tool would then support NCRS staff in the stock management and distribution of *bespoke* NFRI packages that better fit the needs of crisis-affected communities.

Whilst the NFRI Predict model operates at a single household level, the web interface was designed to produce predictions and visualisation across multiple households, to better support NCRS NFRI planning at community and regional levels. Consequently, a middle application layer (hosted on a web server) is required to consolidate requests to and responses from the NFRI Predict model. This application layer - via the web interface - could then also manage a database of saved households for use in future multiple household queries. Henceforth, 'Web Application' refers to the combined application and web interface.

The NFRI Predict Model

The NFRI Predict model has already been developed as an independent part of the project, and the full documentation is available in a public repository (GitHub). This document describes the Web Application, which would manage requests to the CCI ML model and predictions returned from the model.

Model Inputs

The model, accepts the following inputs (the geographic, demographic, health-related, and cultural features of a household), which are reflected in the Adobe XD UI prototype for the proposed Web Application design:

- District

- Percent of the household that are non-male
- Household size
- Whether there are children under 5
- Whether there are residents with health difficulties
- House construction material
- Income Generation Ratio

Product Overview

Figure 1 below shows the entire product including the NFRI Predict model, web server-hosted application layer, and web Interface.

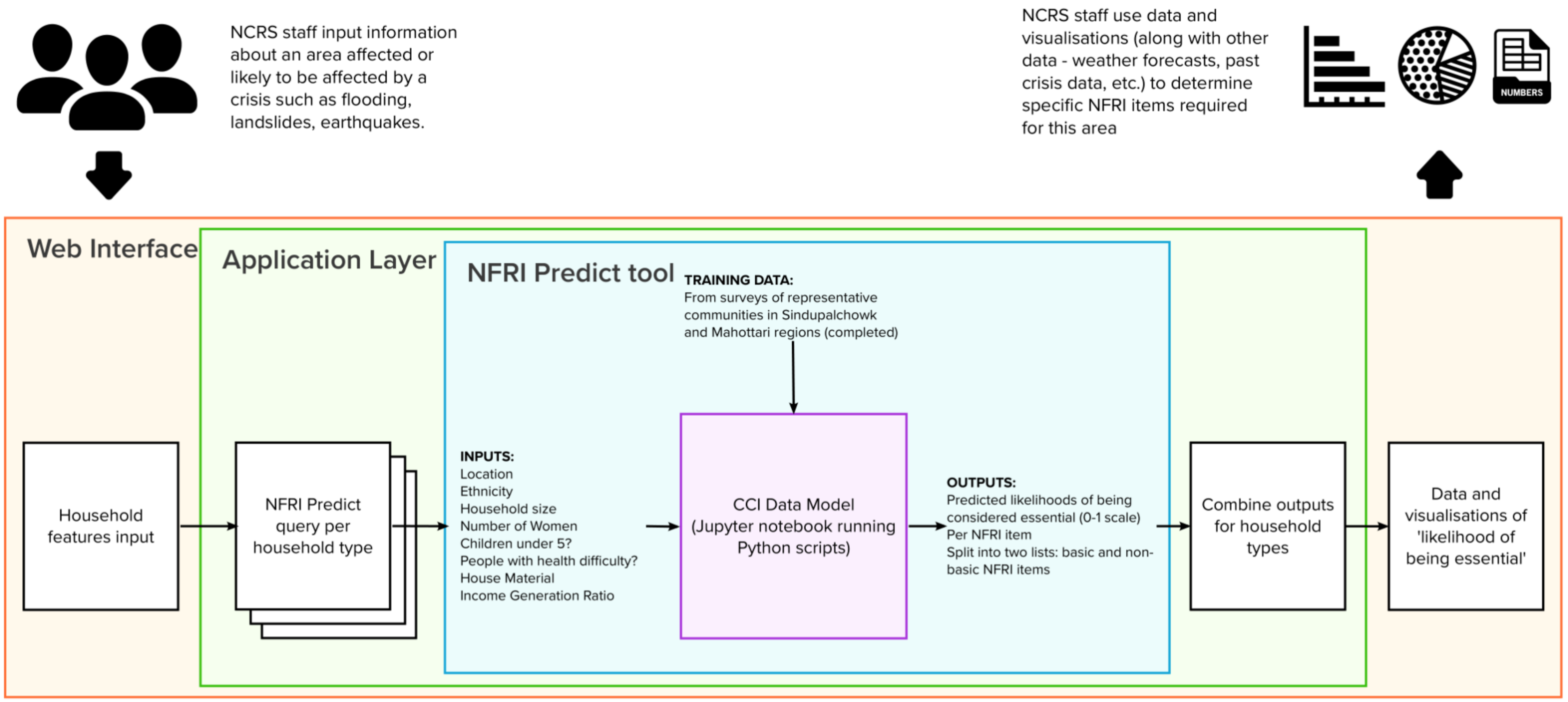

NCRS staff input information about an area affected or likely to be affected by a crisis such as flooding, landslides, earthquakes.
NCRS staff use data and visualisations (along with other data - weather forecasts, past crisis data, etc.) to determine specific NFRI items required for this area
NUMBERS
Web Interface
Application Layer
NFRI Predict tool
TRAINING DATA:
From surveys of representative communities in Sindupalchowk and Mahottari regions (completed)
Household features input
NFRI Predict query per household type
INPUTS:
Location
Ethnicity
Household size
Number of Women
Children under 5?
People with health difficulty?
House Material
Income Generation Ratio
CCI Data Model (Jupyter notebook running Python scripts)
OUTPUTS:
Predicted likelihoods of being considered essential (0-1 scale)
Per NFRI item
Split into two lists: basic and non-basic NFRI items
Combine outputs for household types
Data and visualisations of 'likelihood of being essential'

Figure 1: Product Overview

Expected usage characteristics

Use case	**Expectation**
Create a new prediction by submitting one or more new households, or retrieving one or more saved households, to identify the relative importance of NFRI items for these households.	NRCS staff (e.g. Planning and Monitoring Team) enter features for each new household and/or retrieve one or more saved households to be included in the prediction. (Each new household is named and saved for use in other predictions.) They then receive data and visualisations of the predicted 'likelihood of being considered essential' for each NFRI item (grouped as either basic or non-basic items) for this combination of one or more households.
Load a saved prediction.	NRCS staff select a previously saved prediction for one or more previously entered households then receive data and visualisations as above.
Visualise predictions.	Once the model returns predictions (as above), NCRS staff select to display them as bar charts, heatmaps, or tables of numerical data.
Output predictions.	Once the model returns predictions (as above), NCRS staff select whether to print visualisations and/or tabulated data, and whether in colour or black and white. (User research identified that paper documents are regularly used in planning processes by the NCRS and that black and white printing is the norm.)
Save a prediction.	Once the model returns predictions (as above), NCRS staff save it with a unique name for future use.

Web Application System Architecture

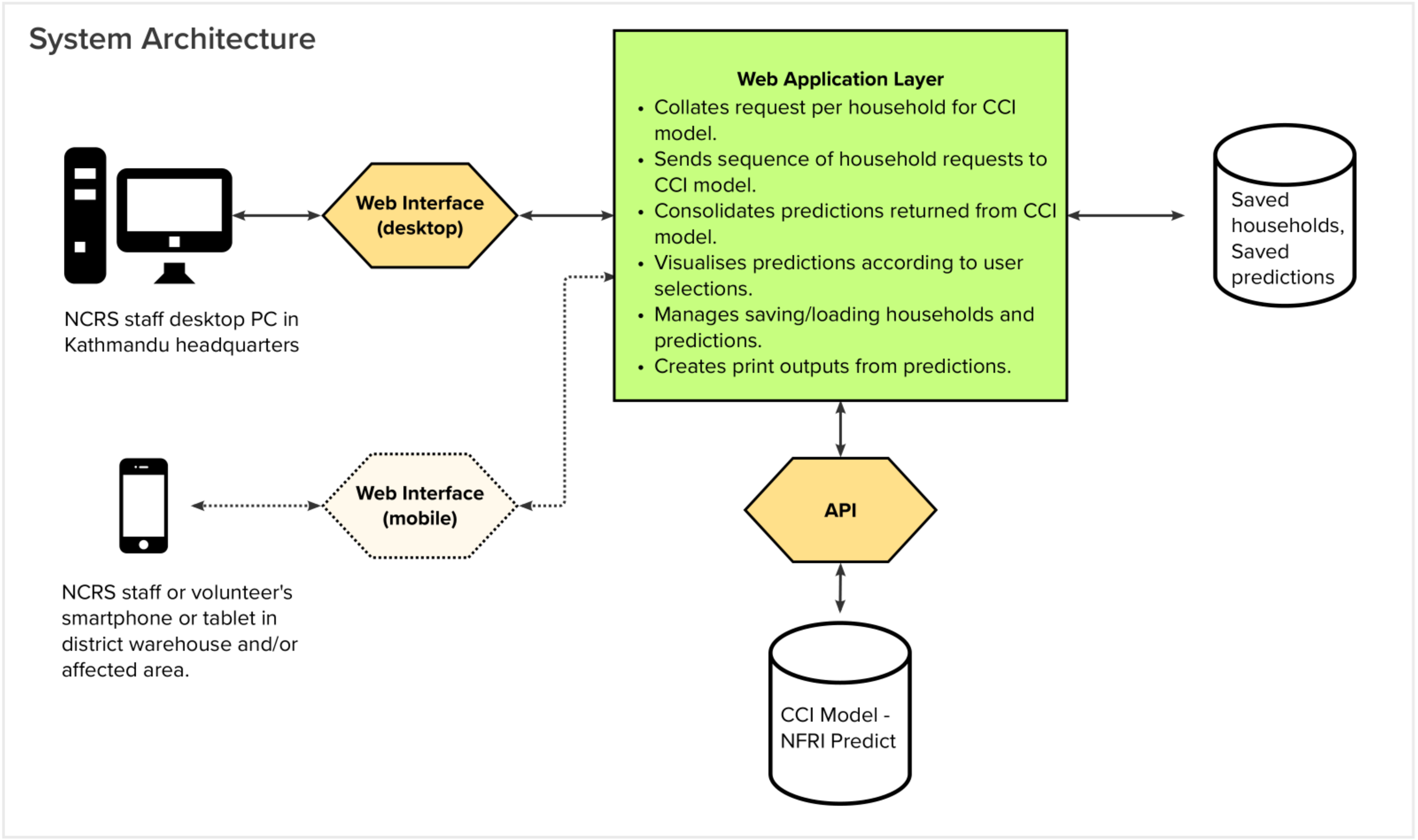

System Architecture
Web Application Layer
• Collates request per household for CCI model.
• Sends sequence of household requests to CCI model.
• Consolidates predictions returned from CCI model.
• Visualises predictions according to user selections.
• Manages saving/loading households and predictions.
• Creates print outputs from predictions.
Web Interface (desktop)
NCRS staff desktop PC in Kathmandu headquarters
Saved households, Saved predictions
Web Interface (mobile)
NCRS staff or volunteer's smartphone or tablet in district warehouse and/or affected area.
API
CCI Model - NFRI Predict

Figure 2: NFRI Predict Web Application system architecture

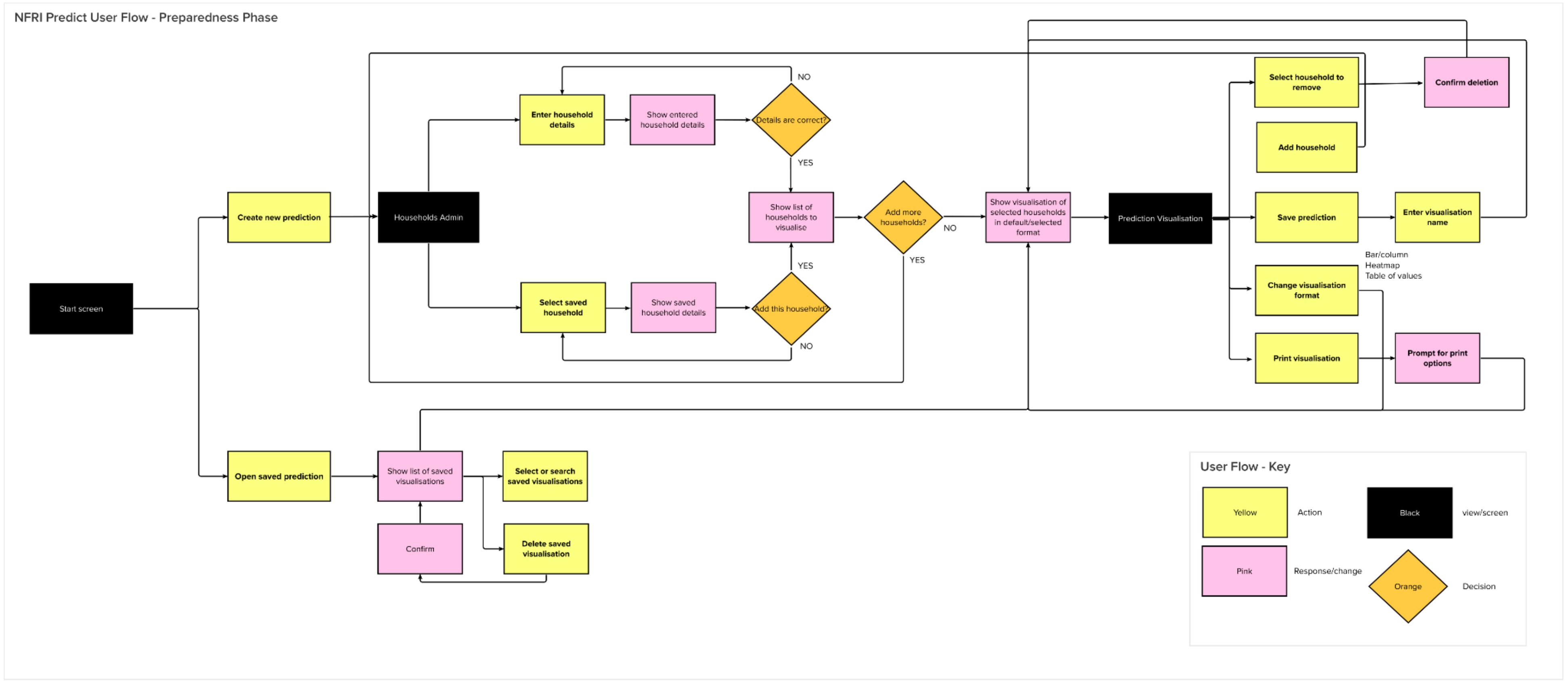
Breakdown of each component:

Component	**Details**
Web Interface (desktop)	Enables the input of household features for generating NFRI predictions, and the control of how prediction outputs are visualised, saved and printed. The UI prototype below illustrates the potential operation of this Web Interface as accessed via a desktop PC at the typical display resolution used by NCRS. Whilst some initial usability testing has been done with this prototype, we recommend further user research to understand how the interface would integrate into NCRS typical working practices and office environments and further design refinements to the UI accordingly.
Application Layer	Required to collate requests, sequence and send requests, and consolidate returned predictions to/from the CCI Model (which operates on a single household basis). Visualises returned predictions either using open source libraries such as D3.js or via an API with visualisation tools such as, again, D3 or Flourish. The visualisations in the UI Prototype below were generated using the Flourish web tool. Manages the saving of household data (features associated with a household, see CCI Model Inputs) and predictions returned, for later use. Creates print-friendly PDFs of returned predictions - visualisations and data.
API	API that directly interacts with the CCI model using standardised inputs and outputs. This will allow the middleware to query the model and receive returned predictions.
CCI Model	Machine learning model developed to predict the 'likelihood of being essential' for individual NFRI items for a given household, see System Overview.
Saved households, saved predictions	A database for storing saved households and saved predictions, which the application layer writes to and reads from.
Web Interface (mobile)	This component was not specified or designed within the project but is a possible extension. User research identified that NCRS staff in district warehouses (where NFRI items are stocked) and NCRS staff and local volunteers in crisis affected areas might also make use of NFRI Predict. For example, to provide feedback on the

	number of NFRI items already stocked and to provide feedback on the accuracy and relevance of predictions. Developing this component would require further UX research and design to identify relevant use cases.

User Flows

The diagram below shows the use of the Web Application by a member of NCRS to inform NFRI planning and stock management during the preparedness phase of emergency management.



UI Prototypes

A prototype of the Web Interface was built in April 2022 with Adobe XD that shows the whole UI and all key interactions.

Links to prototype files	
XD share link	https://xd.adobe.com/view/363a08a4-41a6-4edb-b747-c20a46182709-6369/?fullscreen
XD File	https://github.com/nestauk/cci_nepal

Proposed implementation

We propose the NFRI Predict Web Application is a web application running on a modern Javascript framework to provide the frontend, that is hosted on a Node.Js server backend. An API is provided to allow the client to GET and POST data to an SQL database. In the backend, the SQL database will also receive data output from the model, which are used in the displaying and visualising predictions.

See the system architecture diagram in the introduction section.

System Component Details

The following sections provide detailed descriptions of how each component should function, including technology recommendations.

Web Application

We propose the main web application should use a modern Javascript framework to provide the client interface to the NFRI Predict model. Due to the simple functionality of the web application (sending and receiving from an API, visualising receiving data, etc), we propose the use of Vue.js. To support Vue, the server would run Node.js and the associated packages. See the flexible architecture boxout below for further details on the frontend implementation.

There are three main uses for the web application:

1. Collating and sequentially submitting requests to the NFRI Predict model
2. Consolidating and visualising predictions returned from the NFRI Predict model
3. Managing the saving and loading of household data and predictions data for future use.

A full rundown of the user flow through the app can be seen on the previous pages (here), and an interactive prototype showing all the screens and the user flow can be seen here.

Flexible architecture point: The exact frontend framework used for the web app is flexible. The app itself is relatively simple so does not require complicated functionality. For developer speed we have suggested using a modern Javascript framework (Vue), which has good compatibility on devices. However further work is required to find out exactly which browsers are used by the volunteers and RC Staff, which will then guide the exact decision of the frontend framework.

1. Collating/submitting requests

The NFRI Predict model accepts inputs in the form of features for a single household (see here). However users are likely to wish to see predictions for several households in planning NFRI distribution and stock management. Therefore, the web application will collate requests for predictions for single households that are then sequentially passed, via the API, to the NFRI Predict CCI model.

2. Consolidating and visualising predictions

The NFRI Predict model returns predictions in the form of ‘likelihood of being essential’ for individual NFRI items for the single inputted household. Therefore, the web application will consolidate the multiple returned predictions, according to the number of households sequentially submitted, and then present and visualise these predictions via the web interface.

3. Managing saved households and predictions

NRCS staff will need to save features inputted for certain households for future use in creating predictions with different combinations of households. Similarly, NRCS staff will need to save any predictions generated for later recall. The web application therefore needs to manage the writing and reading of this information to the separate SQL database.

NFRI Predict Model API

The model produces a single output - a set of predicted ‘likelihoods’ for individual NFRI items. We propose the following implementation:

The model is hosted on the same web server that the web application is running on, and an API is written specifically for the model (the *Model API*). The web application and CCI model can then call the Model API, which will process the data passed to it. It would have two endpoints:

1. Process household prediction
    a. This endpoint will be invoked for individual households in a set of collated households in a requested prediction until no households remain. It will pass the household as a set of features data (as per the model inputs).
2. Return household prediction
    a. This endpoint will be invoked by the CCI model once a prediction has been produced. It will pass a prediction as a set of ‘likelihood’ values (0-1) for 11 basic NFRI items and 11 non-basic NFRI items.

Endpoint	**Method**	**Input query/params**	**Expected output**
[Input]	POST	Location Household size Number of Women Children under 5? People with health difficulty? House Material Income Generation Ratio	
[output]	GET		22 x “Likelihood’ values 0-1, for: 11 basic NFRI items 11 non-basic NFRI items

Database Structure

The following table is a proposed database structure to support the proposed system architecture. There is some relation between the tables (e.g. predictions consisting of several saved households), so we suggest a relational SQL database be used. The functionality is simple, so any mainstream implementation of an SQL database would be suitable, such as PostgreSQL.

Table	Details
Households	Feature values for a single household, plus user-defined name.
Predictions	One or more saved households - as above.

Database API

The NFRI Predict Web Application is primarily focussed on the generation and presentation of prediction visualisations. Whilst this process does not depend on the database, managing saved households and predictions simplifies the process of generating predictions through reducing the amount of data users must input (i.e. for frequently used households). The Database API handles requests for the database, primarily fetching data for the web application, and handling input from the web application. A proposed API specification is provided below:

Endpoint	**Method**	**Input query/params**	**Expected response**	**Onward processing**
[input]	POST	1. Household name 2. Household features	HTTP code 200	Stored in Database
[input]	POST	1. Prediction name 2 .Household names	HTTP code 200	Stored in Database
[output]	GET	Household name	Household features	-
[output]	GET	Prediction name	Household names	-

Security considerations

The current implementation does not implement a security layer. Users of the NFRI Predict Web Application are able to view and enter data about specific households. Whilst this data does not include names or other information that could be used to personally identify individuals, the identity of household members might be inferred from the features of their household. Therefore, it is important to implement some access control to stop unauthorised usage. Being a web application, NFRI Predict will be available to everyone with an internet connection. Security through obfuscation (e.g. not advertising the service, keeping the link secret, etc) will not be sufficient, as once an unauthorised user finds the service they can provide direct input to the system.

The simplest solution would be to implement a username and password system. This would require the username and password details to be input before use. This will stop any unauthorised usage, but would require some administrative work, either from the developers or from RC staff. For example someone in RC could be tasked with providing usernames and passwords to new volunteers/staff who wanted to use the tool.

Beyond access control, the APIs should implement SSL and therefore would be encrypted between the client and the server. There is no personal data used in household features, therefore a simple username and password system should provide sufficient protection.

Projected Time & Costings

Assumptions: All development time is full-time, based on one developer working 9am - 5pm. Time estimates are conservative to factor in unforeseen technical challenges.
Financial cost assumed in GBP

#	Task / Item	Details	Time cost	Financial cost
1	Developing web application, web interface	Writing frontend code, implementing UI designs, send/receive appropriate data for display.	6 weeks	Developer time x time cost
2	Developing web application, backend (inc APIs)	Configuring Node.Js server, writing database API, configuring to receive model input.	3 weeks	“ “
3	Configuring resources to host system	Configuring server, setting up databases, configuring send/receive of data between APIs.	2 weeks	“ “
4	Alpha Testing	Testing with small group of users (either external or internal to IFRC)	4 weeks	“ “
5	Iteration based on Alpha testing	Fixing bugs and making changes suggested from the alpha testing.	3 weeks	“ “
6	Beta Testing	Testing in Nepal	4 weeks	“ “
7	Iteration based on Alpha testing	Fixing bugs and making changes suggested from the beta testing.	3 weeks	“ “
		Development Total	25 weeks	
	Recurring			
1	Technical support	Bug fixing, maintaining service	Variable	2 months ‘on call’ development time

		availability. Sensible to budget for at least 2 months technical support when the tool is launched to fix issues that arise in the field. Depending on scale of deployment and use of the tool, a service-level agreement should be discussed for expected uptime and maintenance response times.		once tool is released Service-level agreement TBC