SPECIFICATION

DOCUMENT NUMBER	:	R2020_06_21_2
DOCUMENT TYPE	:	SOFTWARE REQUIREMENT SPECIFICATION (SRS)
VERSION	:	1.6
SOURCE	:	UNICEF
PREPARED BY	:	KAZ SOFTWARE LTD.
DOCUMENT DATE	:	11/07/2020
LAST UPDATED	:	28/07/2020
PROJECT TITLE	:	EDUCATION SECTOR INFORMATION MANAGEMENT SYSTEM

1. Introduction

The Software Requirement Specification Document (SRS Document) is a written statement of something that can be done and that cannot be done by software. The document is only acting as a foundation or basis for an agreement between the parties-party software developers with the software users. The purpose of this document is to outline the requirements for UNICEF and Education Sector Partners with a Managed Information System. Different sections of this document will explain different feature and business requirement that UNICEF raised, and Kaz Software will try to provide solutions to.

1.1. Purpose of the project

Kaz Software ltd. is required to develop/adopt a customizable web-based system for efficient information management that meets the sector needs. These needs include tracking status of enrollment of children by yearly, level/grade, teacher training, status of learning spaces (damage tracking) and community engagement and participation. The Sector requires to have on disaggregated reports of children per demographic attributes such as age, disability, and sex.

Based on the JRP indicators the system will enable tracking and reporting on various indicators across the objectives on access to education services, quality, and community engagement. The reports include geo-mapping of the different attributes as per the sectoral monitoring framework inclusive facilities mapping per status and type. This will also include gap analysis of areas without education facilities and displayed on the geo map.

To enable data-driven decision making by the education sector partners, the system will enhance storage, analysis, reporting for the education sector data. The system will also be required to enable reporting through visualization dashboard with GIS location of the learning facilities and funding status of agencies and implementing partners.

2. Overall Description

2.1. Business Requirement

The 2019 Education Sector Joint Response Plan (JRP) estimates that 462,400 children and youth aged 3-24 years old need learning and education services. Since December of 2019, over 324,367 refugee children aged between 3 to 24 years of age has access to education through the 5,495 learning facilities established across the congested camps by over 30 education sector partners. All partners supporting education Program report enrolment in various learning facilities and related data to the education sector for storage, analysis, and dissemination. These reports are then consolidated in a final report referred as the 5W tool. The filling of this tool is done manual work and there are also the integrity and reliability of the data is diminished due to the large size of the data and number of data items collected.

This causes the following key gaps in the current system:

- **Lack of validation check**
- **Activity logging**
- **Data security**
- **Efficiency in data collection, analysis, and reporting**
- **Data visualization**
- **Ghost and duplicate records**

The education sector database is the official source of data for tracking progress of the education response. It is therefore critical that the data collected and published by the education sector is as reliable. The education sector requires technical support from a third-party agency to address the technical gaps of the current system and improve integrity of the education sector information management system.

2.2. Product Features

Kaz Software will develop a customizable web-based application system that will allow users to efficiently manage information that meets their need. These needs include tracking status of beneficiary enrolment of children in periodical, level/grade, teacher training, status of learning facilities that they run, engagement and participation.

The application should allow data driven decision making by the education sector partners, the system will enhance storage, analysis, reporting for the education sector data. The system will also be required to enable reporting through visualization dashboard with location coordinates of the learning facilities and funding status of agencies and implementation partners.

2.3. Stakeholders

UNICEF

Name	**Role**	**Email**	**Phone**
Muntashir Arefin	Primary Project Contact	marefin@unicef.org	+8801521505103
Charles Avelino	Education Manager	cavelino@unicef.org	
Brenda Gunde	ICT Specialist	bgunde@unicef.org	
Sharmila Pillai		spillai@unicef.org	
Pierre Lorioux	Information Management Officer	plorioux@unicef.org	
Mahmudul Hasan	PMR	mahmhasan@unicef.org	

Kaz Software

Name	Role	Email	Phone
Mosfiqur Rahman	Project Manager & Primary Contact	mosfiqur@gmail.com	+8801716130287
Nur A Shawal	Chief Technology Officer	shawal.siddique@kaz.com.bd	+8801617607518
Md. Asef Fakih	SQA Lead	asef.fakih@gmail.com	+8801675442145
Tahsin Zaman	Product Manager	zamanw100@gmail.com	+8801974645404

Education Sector Partners List

Education Sector Partners	Abbreviation
Action contre la Faim	ACF
Allama Fazlulla Foundation	AFF
Ananda Magra Universal Relief Team	AMURT
Social Welfare and Development Organization	SKUS
Bangladesh Rural Advancement Committee	BRAC
Caritas Bangladesh	CB
Compassion International Bangladesh	CI
Coastal Association For Social Transformation Trust	COAST
Community Development Centre	CODEC
Dhaka Ahsania Mission	DAM
Danish Church Aid	DCA
Education Development and Services	EDAS
Friendship	
Handicap International	HI
Health Management BD Foundation	HMBD
International rescue Committee	IRC
Integrated Social Development Effort	ISDE
Jagorani Chakra Foundation	JCF
Muslim Hands International	MHI
MUKTI	
Programme for Helpless and Lagged Societies	PHALS
Plan International	PLAN
Prantic Unnayan Society	Prantic
RISDA Bangladesh	
Reaching People in Need	RPN
Shehora Bohumukhi Samaj Kollyan Samity	SBSKS
Save the Children International	SCI
Technical Assistance Inc	TAI

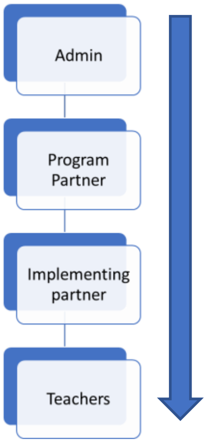
Thailand BRAC	TAI-BRAC
United Nations High Commission for Refugees	UNHCR
United Nations Childrens' Fund	UNICEF
Unite Theatre for Social Action	UTSA
Voluntary Services Overseas	VSO
World Vision International	WVI
Young Power in Social Action	YPSA
Association of Muslim Advancement Network	AMAN
Center for Disability in Development	CCD
Room to Read	RTR
The United Nations Educational, Scientific and Cultural Organization	UNESCO
EDUCO	
	CBM
World Food Programme	WFP

2.4. System user classification

2.4.1. Level

Level defines the hierarchy of users (actors) for the system. Level will be set against users. There are four level of users, from top to bottom:

i. ***Admins***: Users who will be able to manage every other user from any level.
ii. ***Program Partners***: Users who will only be able to manage users from levels below them.
iii. ***Implementing Partners***: Users who will only be able to manage users from level below them.
iv. ***Teachers***: User who will not be able to manage any other users from any level.



2.4.2. Roles

Roles are pre-defined with Permission sets. Permissions are different ability to do actions. Each permission is an action. Permissions are system defined and cannot be altered. The pre-defined roles:

i. Admins will have all permissions.
ii. Program Partner will have the permissions to view, edit, manage users and data below their level, i.e. Implementing Partner, Teacher.
iii. Implementing Partner will have the permissions to view, edit, manage users and data below their level, i.e. Teacher.
iv. Teacher will only able to collect data and provide that data to levels above them.

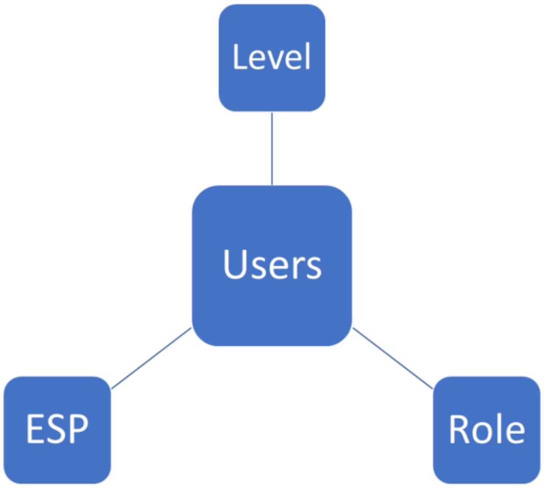
Roles are set against users. Roles can be customizable. Users can create new roles and customize their permission set.

2.4.3. ESP

Education Sector Partner are organization that are represented as Program Partners and Implementing Partners to UNICEF. ESP partners will be assigned against users and facilities. ESP are also stakeholders of the system. Every user will have at least one ESP and maximum two ESPs.

2.4.4. Users

Users are the person who has access to the system. Users' level of access and availability will be defined by the combination of Level, Role, and ESP.



Different users with different roles and levels and ESPs will have different types of data accessibility in the system.

2.5. Dependencies

All requirements and features explained in this document was described and designed based on the discussion between Kaz Software stakeholders and UNICEF stakeholders. These features described will be developed using the discussion points between stakeholders. However, for features like reporting, dashboard, and export, will be developed based on sample structures and templates provided by UNICEF stakeholders. Without the appropriate files and samples, system development will be difficult. UNICEF stakeholders will also have to identify and provide the required parameters using which users will be able to use the features. After providing these files and samples, development of the system can be started. All deliverables from UNICEF should be provided before the development starts

Explained in detail in this document in later section. For data to be stored and managed, there will Fixed Columns in the system, these columns will be of six types.

3. Functional Requirements

This section of the SRS document illustrates organizing the functional requirements for the product by system features, the major services provided by the product. The following section is organized by use cases, user classes, or combinations of both. The objective is to make the most logical sense of the product's main functional requirements.

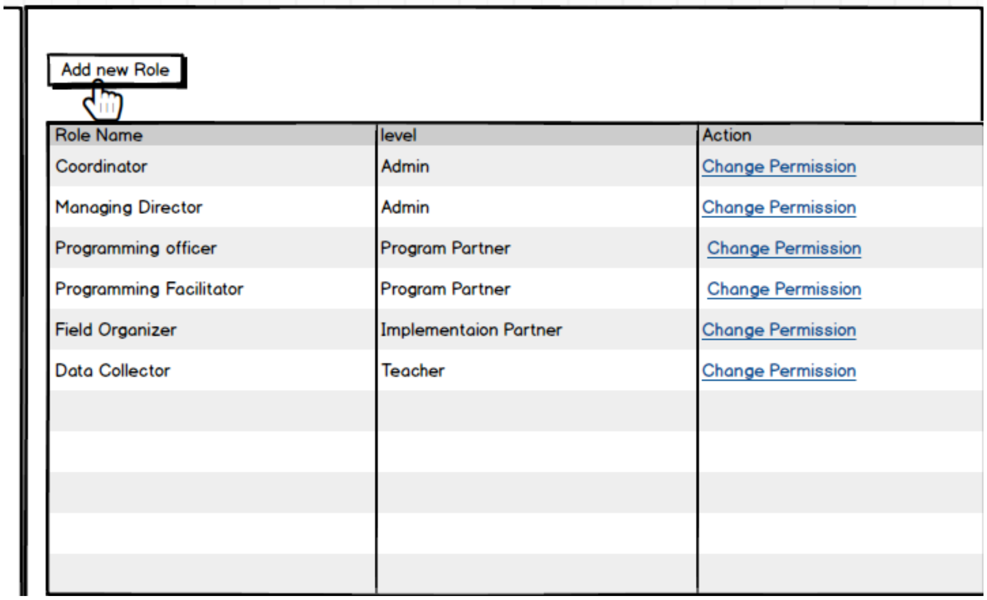
There will be multiple grids for data collection, storage, and reporting. Grids will consist of columns and columns will contain data types. These types will determine the data that is put under each of the columns, as designed by the user. A column can hold only one type of data. The types of columns are:

- **Text:** Only Text data type can be stored.
- **Numeric:** Only Number data can be stored.
- **Decimal:** Only Numbers with decimal figures can be stored.
- **Equation:** Will allow the data stored, can perform arithmetic operations (addition, multiplication, subtraction, division), on the numeric and decimal type fields.
- **Date:** Data type will only consist of dates, e.g. YYYY-MM-DD
- **Boolean:** Data that can be described with only two values: true or false.
- **List:** Data may contain multiple values based on selection from a predefined list, or sequence.

3.1. User, Roles, Permissions

The users will be able to register with their email, access and perform tasks in the application according to their given permission. Each user will be assigned a role which will have its own permission set and each user will be assigned a level. The hierarchy of level is explained in detail in System User classification (see section 2.4).

3.1.1. Role Management grid

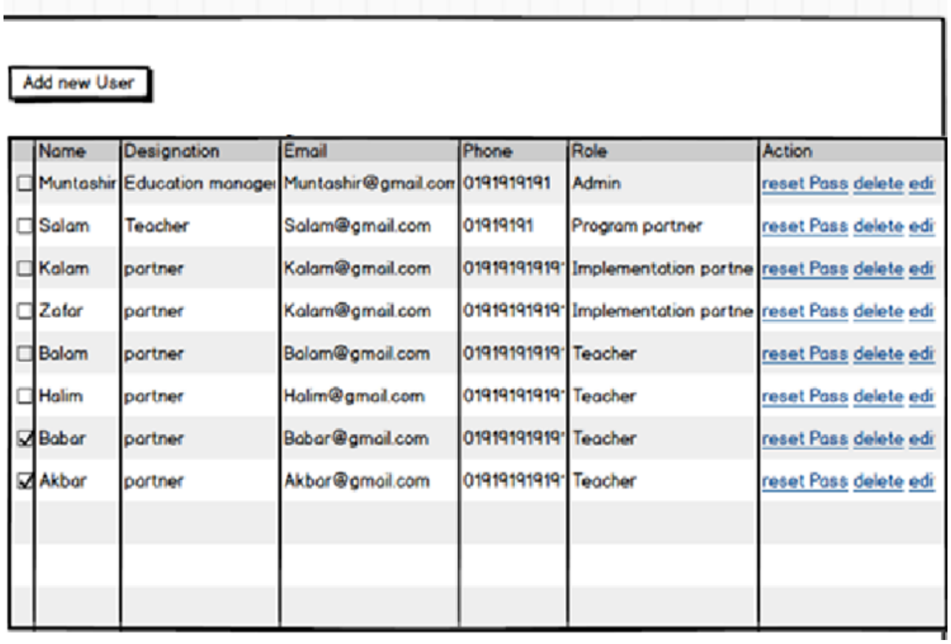


Role Name	level	Action
Coordinator	Admin	Change Permission
Managing Director	Admin	Change Permission
Programming officer	Program Partner	Change Permission
Programming Facilitator	Program Partner	Change Permission
Field Organizer	Implementaion Partner	Change Permission
Data Collector	Teacher	Change Permission

- Permissions are system defined and cannot be altered.
- New role can be added by the users with the add role permission

- After creating the role, the permission set can be altered. The permission set is the granular breakdown of the actions in the application (create, edit, update, delete, approve, view). The user can give any set of permission to any role, but its level will remain the same

3.1.2. User management Grid



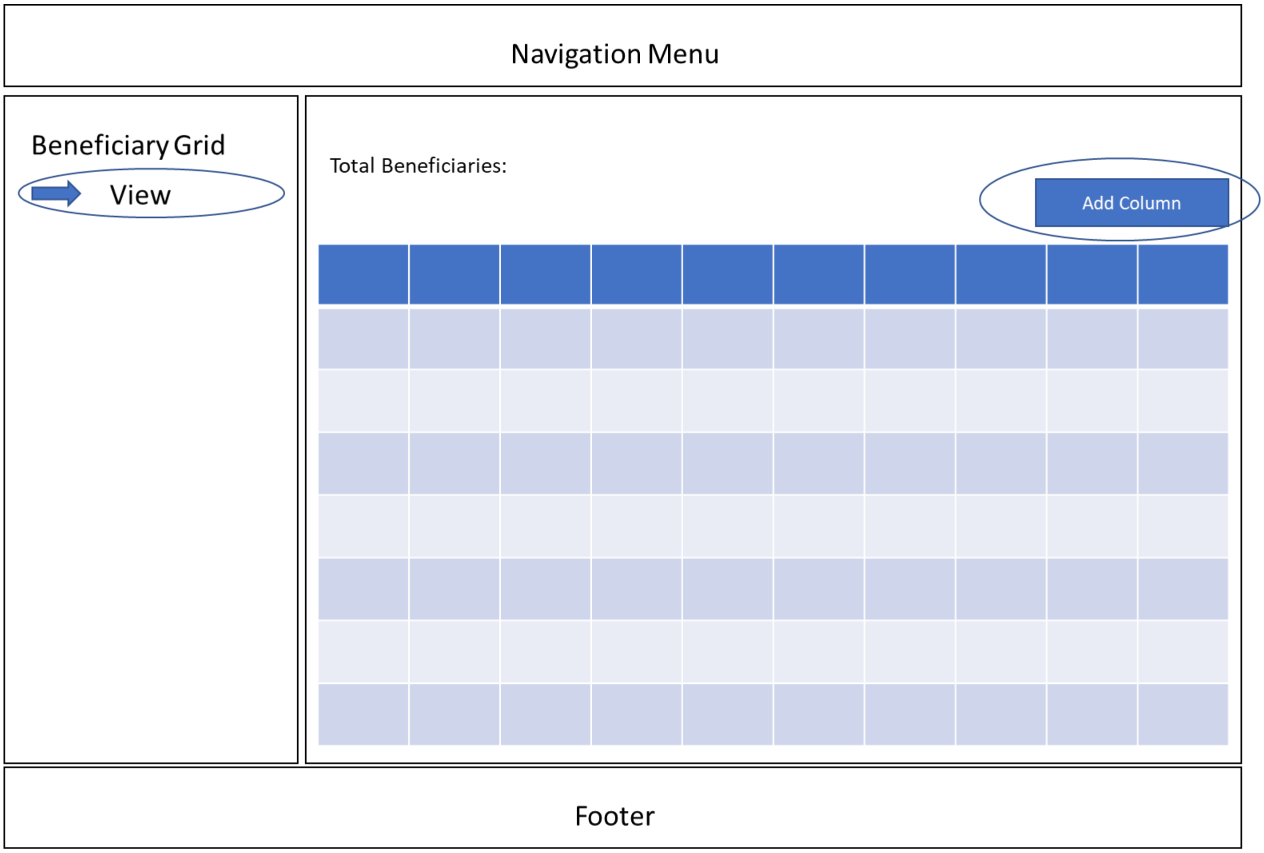
	Name	Designation	Email	Phone	Role	Action
□	Muntashir	Education manager	Muntashir@gmail.com	0191919191	Admin	reset Pass delete edi
□	Salam	Teacher	Salam@gmail.com	01919191	Program partner	reset Pass delete edi
□	Kalam	partner	Kalam@gmail.com	01919191919	Implementation partne	reset Pass delete edi
□	Zafar	partner	Kalam@gmail.com	01919191919	Implementation partne	reset Pass delete edi
□	Balam	partner	Balam@gmail.com	01919191919	Teacher	reset Pass delete edi
□	Halim	partner	Halim@gmail.com	01919191919	Teacher	reset Pass delete edi
☑	Babar	partner	Babar@gmail.com	01919191919	Teacher	reset Pass delete edi
☑	Akbar	partner	Akbar@gmail.com	01919191919	Teacher	reset Pass delete edi

- The user management grid will show the users created for the application who can access the application to perform tasks.
- When a new user is created, he must provide his email and some pre-defined fields (which will be provided by UNICEF to Kaz before the Development starts). While creating the user his role will also be assigned (from role grid). The user level will be defined.
- When a new user is created, he will receive a system generated password in the email and can log in with that password.
- The user can be deleted (the user and his email will be deleted)
- The password can be reset.
- The user fields can be edited
- The user can be assigned a different role. With this the permission set of the user will change with this alteration.
- Any user with the add new user can only add and view users below his level (Admin-> Programming partner –> Implementation Partner -> Teacher)
- Users with beneficiary or facility grid permissions can only see the facility he is assigned to or users from the same level are assigned to.
- In the user grid there will be option to add custom columns
- There will be a filter option by Designation (not role/level) and then export the filtered data.
- Users will who has at least one ESP in common, will be able to view Beneficiary and Facility data assigned to at least one of those users.

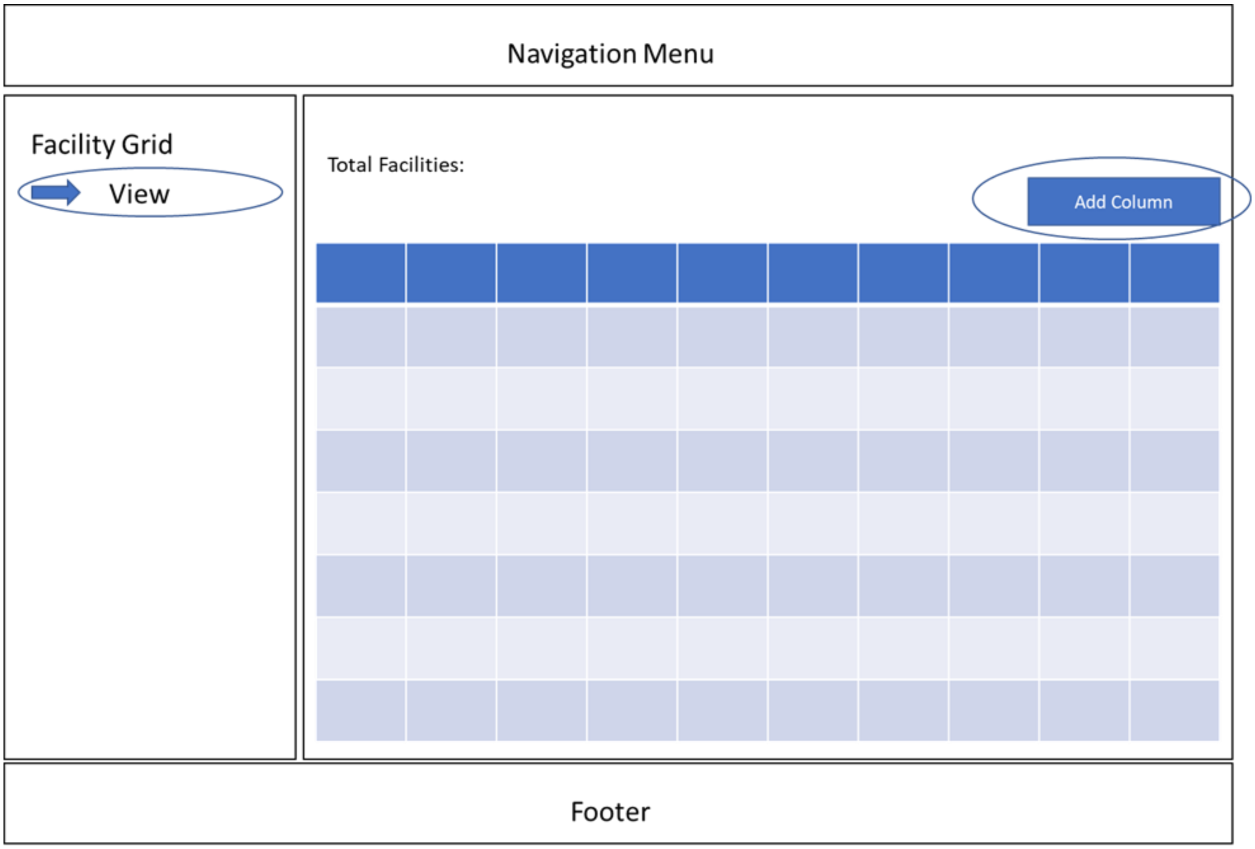
- User management needs to be flexible. User can change roles and permission set based on need.
- A user who is PP ESP in one facility can act as IP ESP in another Facility.
- A user who is PP level in one facility can act as IP level in another facility.

3.2. Columns

In the system, all types of data will be stored in grids. There are multiple grids that are mentioned in the below sections. These grids will be consisting of columns. Columns represent properties of each beneficiary and facility.



Grids will contain both **Fixed Columns** and **Dynamic Columns**. A column represents an indicator. Columns can be used for reporting and dashboard filtering (See sections 3.8 and 3.9). Columns, defined by UNICEF, will available for filter options in reporting and dashboard by default but users can also generate reports and dashboard information by selecting the columns from each grid. Both Fixed columns and Dynamic Columns can be used for filtering in reporting and dashboard.



3.2.1. Fixed Columns

Fixed columns are pre-defined properties by UNICEF. Fixed columns will be used to add new beneficiaries and facilities. Fixed columns will also be used as indicators for monitoring processes (See section 3.5).

Each column will be an indicator that users will be to select while they are creating data collection forms.

Users will not be able to change or delete Fixed Column headers.

Fixed columns will have types:

- Text
- Numeric
- Decimal
- Date
- Boolean
- List

3.2.2. Dynamic Columns

Users will be able to add as much columns to the grid as they require. These are dynamic columns

When adding a new column, the users will have to set the type of column it is:

- Text
- Numeric

- Decimal
- Equation
- Date
- Boolean
- List

Users can add new column either in the general grid of beneficiaries or they can also add a new column in their desired views (See sections 3.3.2 & 3.4.2).

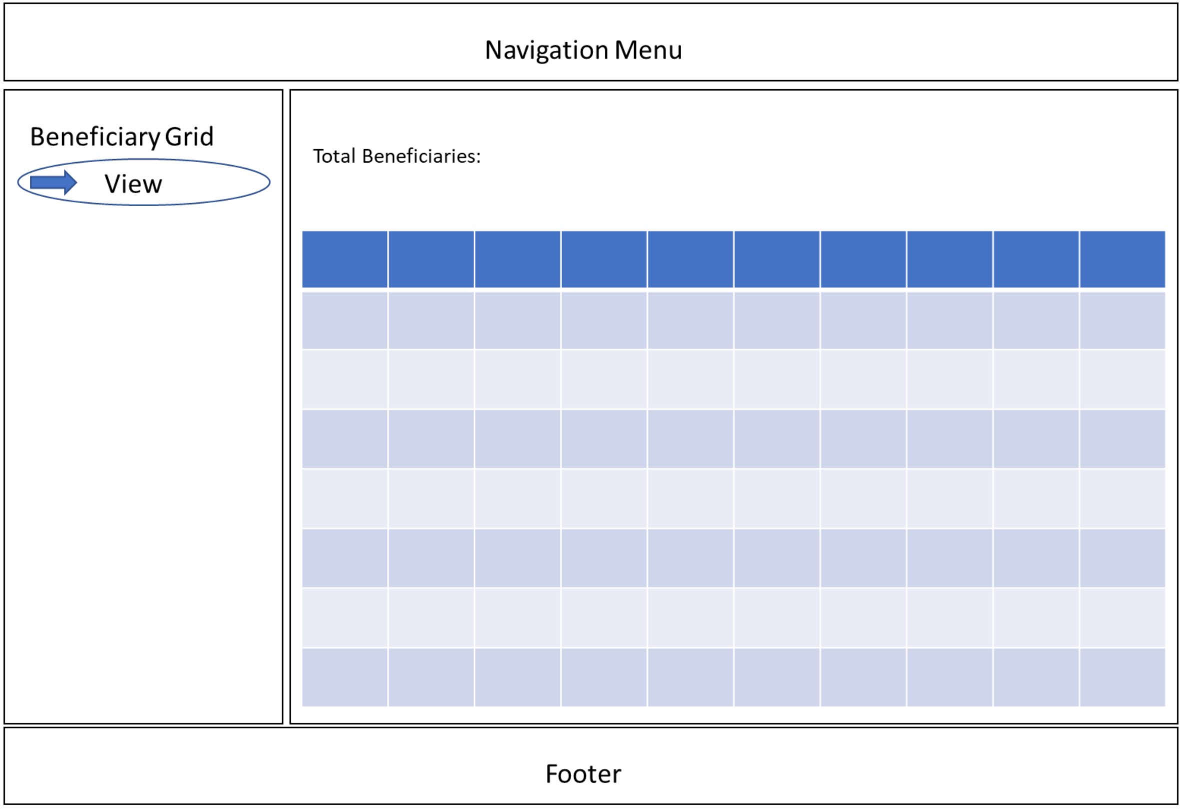
Users can change Dynamic Column headers.

When adding new columns, users will have to provide the max limit for characters input that field while data collection process.

3.3. Beneficiary Data

There is over hundreds of thousand data collected regarding Beneficiaries (Rohingya Children between the age of 3-24 years), and it has different categories, or indicators, or columns. These data are collected once and some of these data are updated over different periods. These data are needed to be shown in a grid view (table view) to represent all the columns of all the beneficiaries that Education sector is working with.

3.3.1. Beneficiary Data Grid



There will be a grid that is dedicated to show all the beneficiaries.

Each column will contain a property for a beneficiary.

When selected from the menu, all the beneficiaries' along with their default columns (*will be confirmed by UNICEF*) must be shown in columns, e.g.: Name, Father's Name, Mother's Name, Age, Gender, Camp, etc.

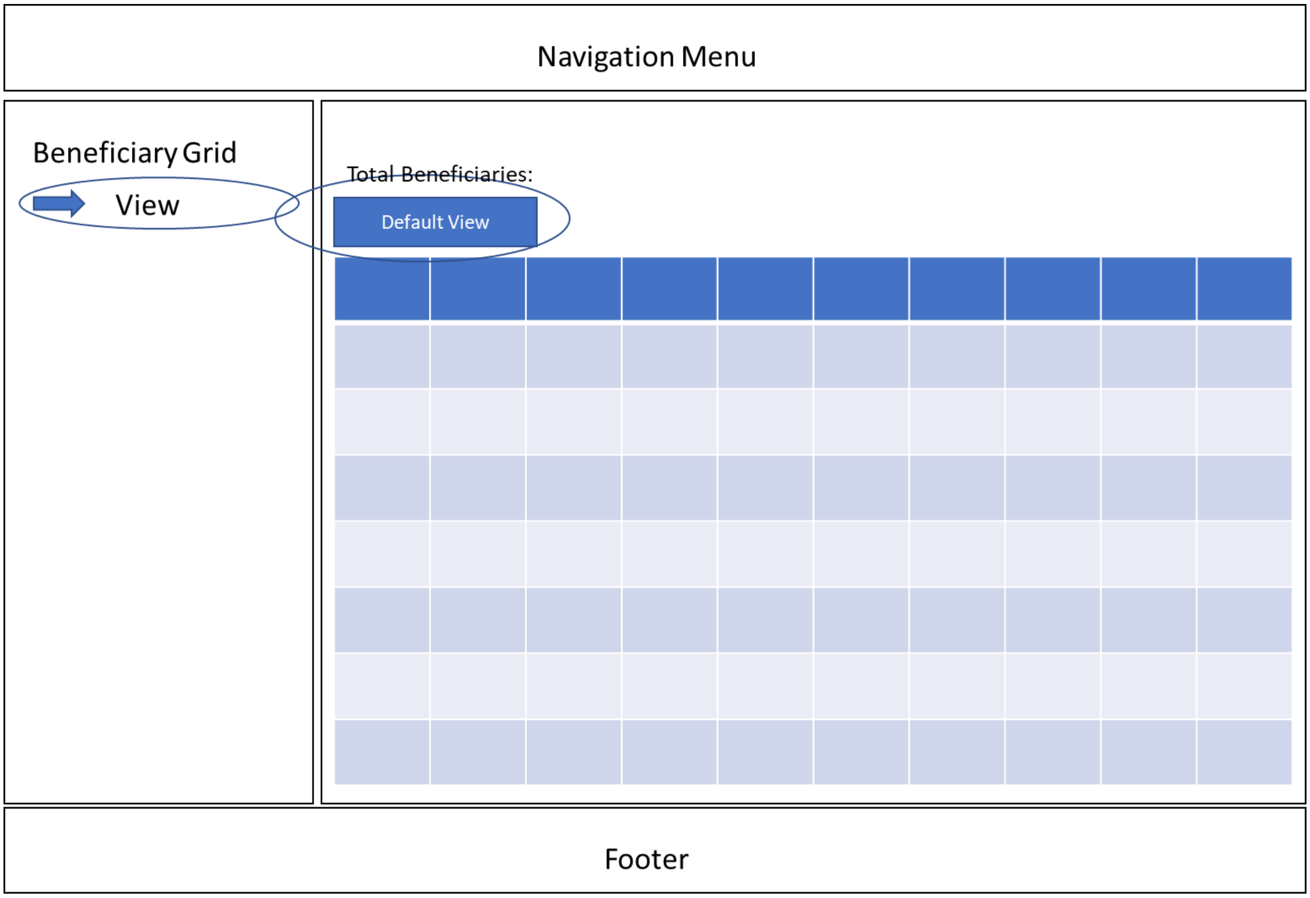
Each column will contain a type and based on those types; data will be stored under those columns. Different types of columns:

- Text
- Numeric
- Decimal
- Equation
- Date
- Boolean
- List

All data collected will be stored under this grid.

On the top, it will show the total number of beneficiaries in the grid view.

3.3.2. Customized Views for Grid Data



In default, the grid will not show all the beneficiaries' basic data, e.g.: Name, Father's Name, Mother's Name, etc. Default view will only show Fixed Columns. *[These basic columns will be defined by the UNICEF]*

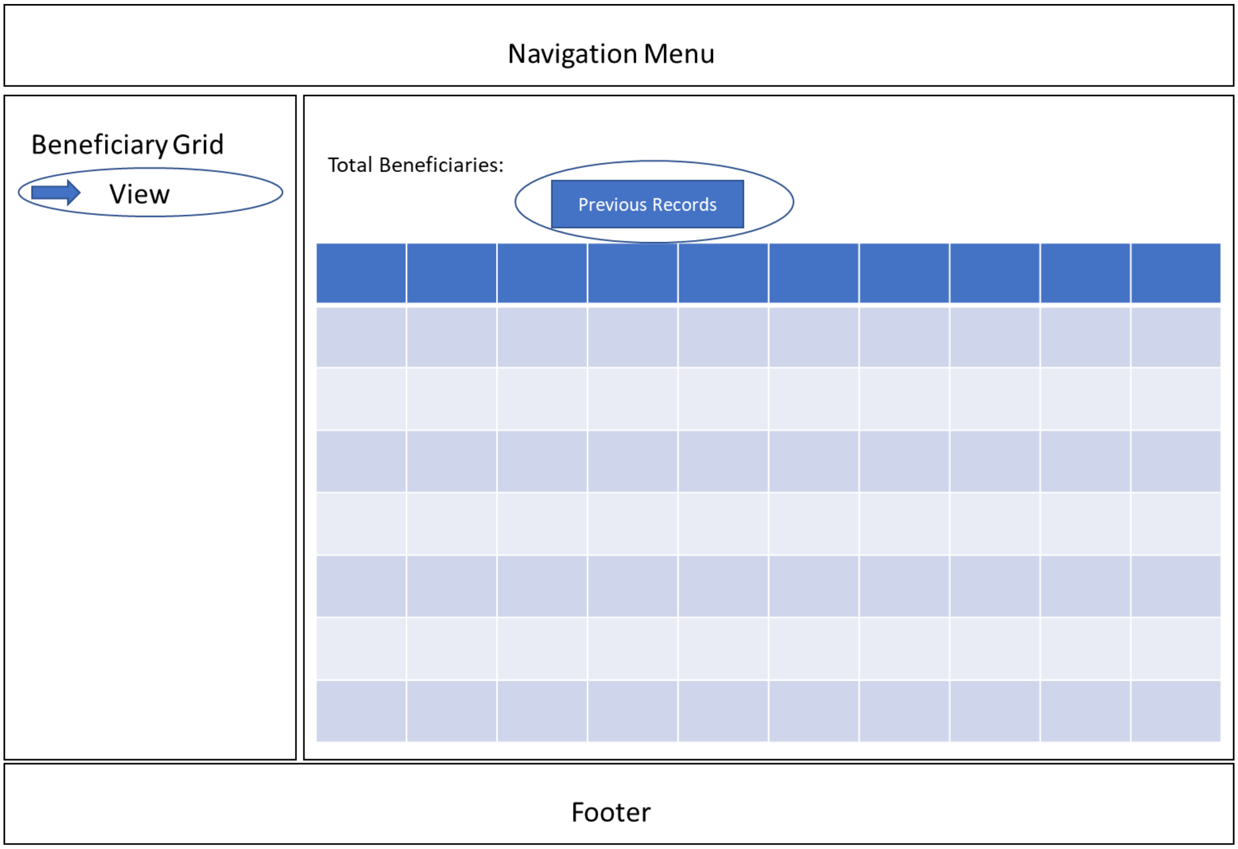
In default view, it will not show all columns for beneficiaries. Default view will only show Fixed Columns.

Users can change the view according to their specific needs.

Users will be able to select the columns from a columns' list and create a new view for the beneficiaries.

Users will also be able to save and set a name for those views.

3.3.3. Store records in different dates



Data stored in the grid can be updated over time.

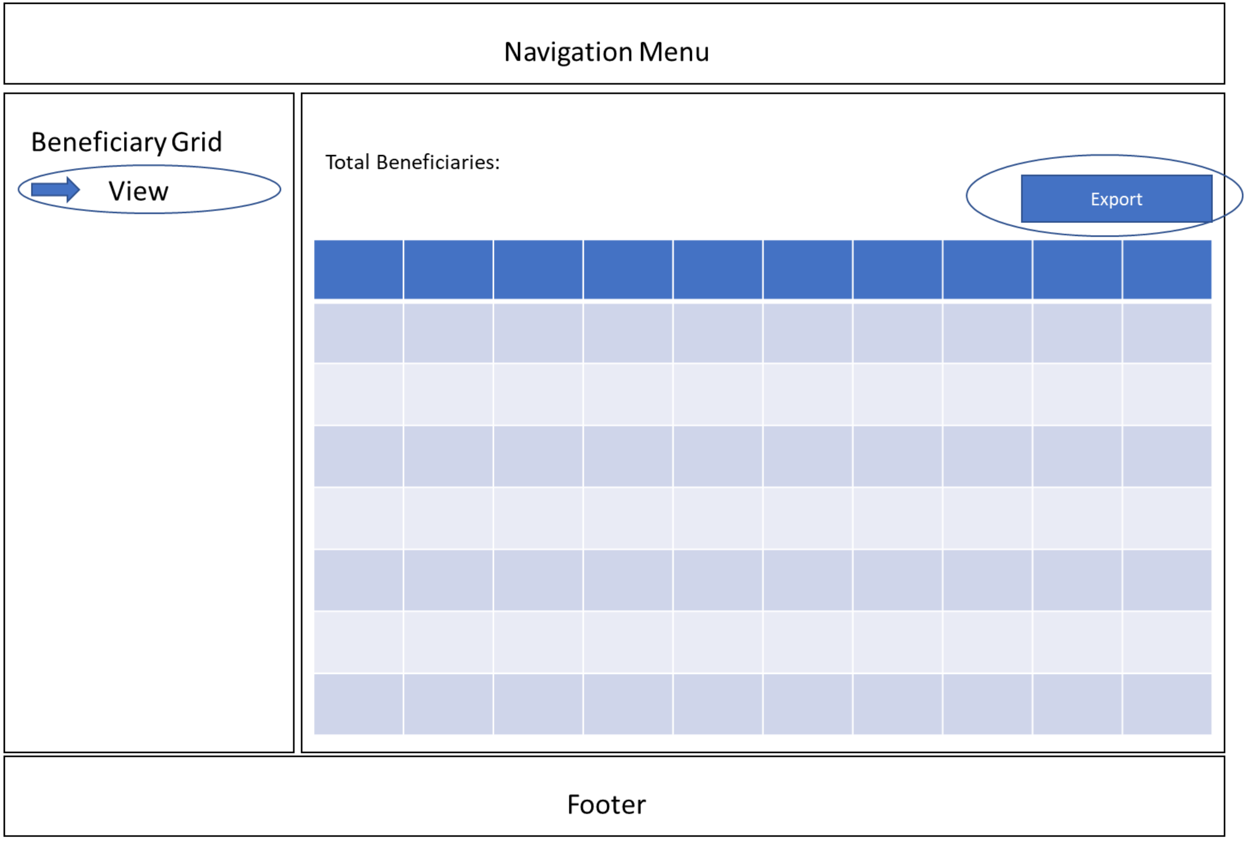
Records will be stored from different collection instances, in different time, using the monitoring section (See section 3.5), users will be able to access those instances.

Users can choose a specific collection instance which will load the data collected from that point of time in the beneficiary grid.

The instances can be searched and then selected. Once the instance is selected the grid will show the data of that collection instance.

By default, in the beneficiary grid, it will always show the latest instance of data.

3.3.4. Export Beneficiary Data

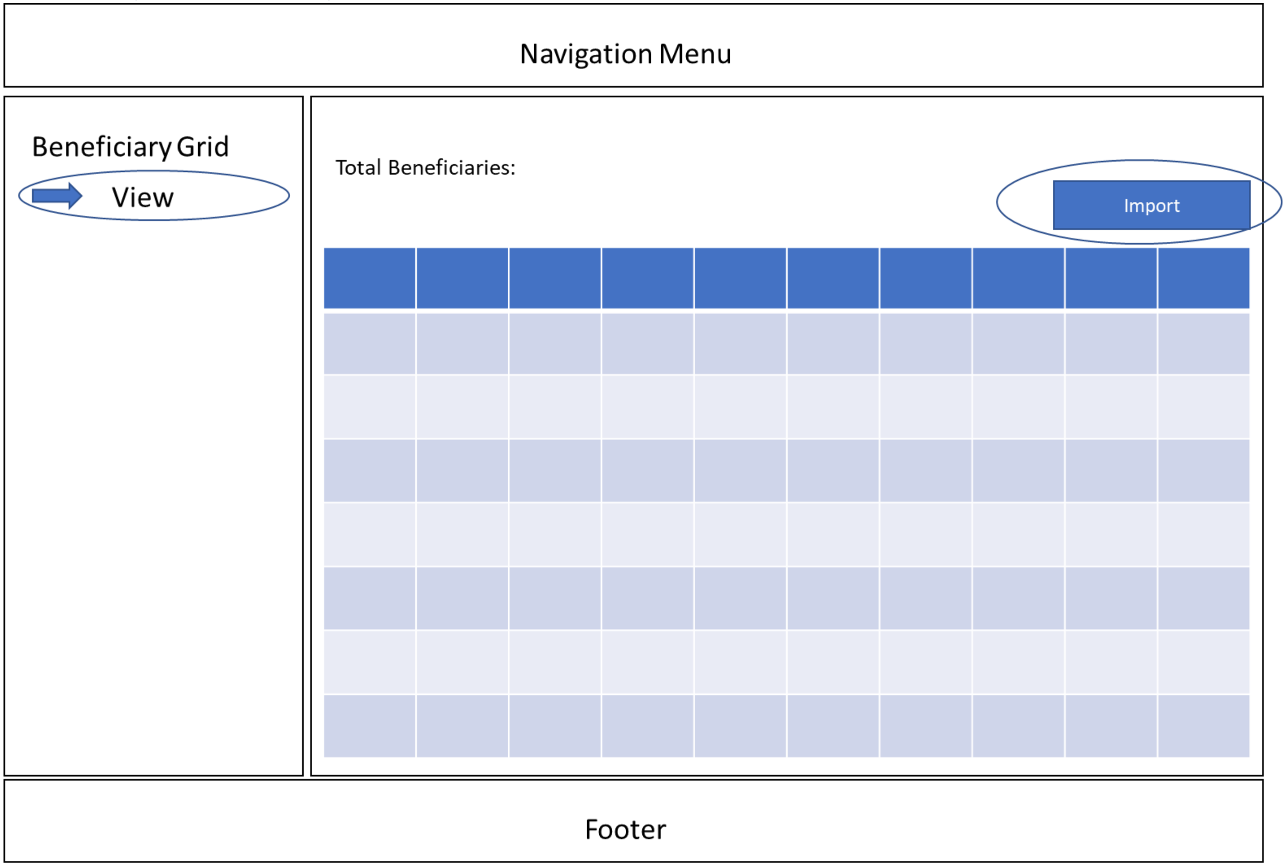


Users will be able to export the data from this grid.

This will export all the columns from the beneficiary grid.

Export will offer excel, csv file format and allow to share on Google Drive.

3.3.5. Import Beneficiary data



Users will be able to import beneficiaries and add them to the grid.

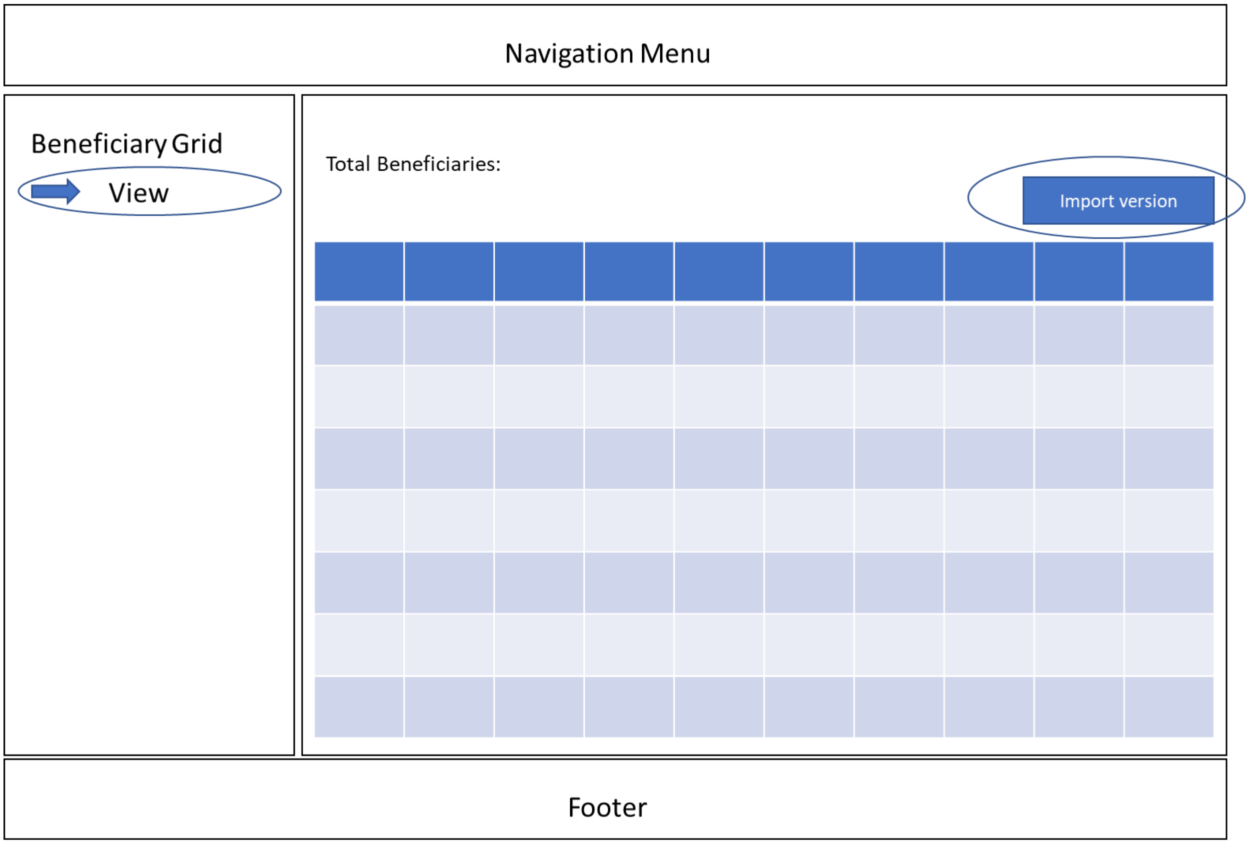
Import from an excel file.

Based on the columns, users will have to create the dynamic columns first, in the grid. This also allow them to import data with dynamic columns.

Import will first download a template. The template will be used arranging the beneficiary data for import.

Import beneficiary data will include data validation to check for erroneous data content in the template file.

While importing, system will run data validation to check for erroneous data. If the system finds error in data while importing, it will show messages to identify where in import file is the error.



There will be an “Import version” feature that will users to import beneficiaries’ data from different version. This can allow users to import data from a previous version.

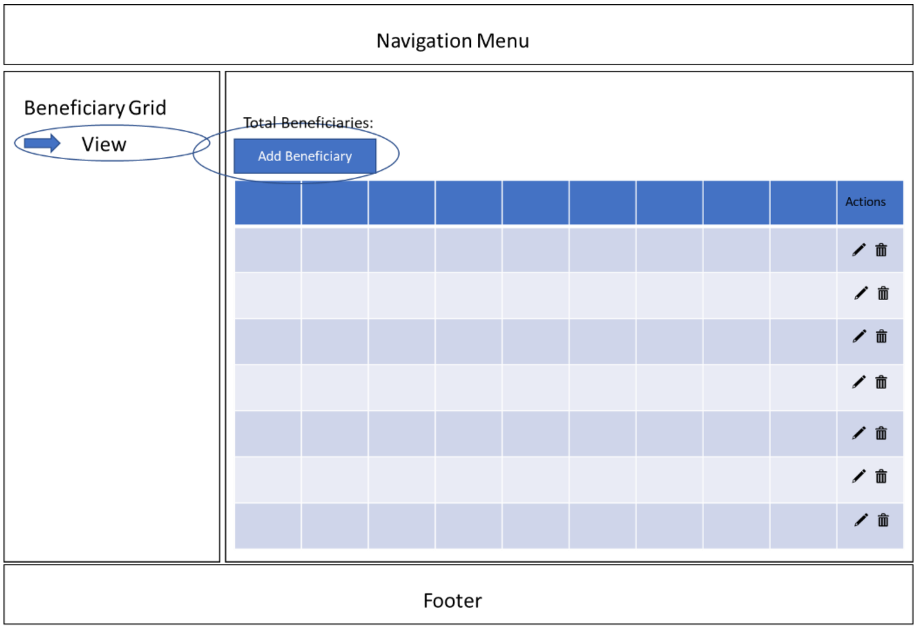
To import previous version of data, users will have to provide:

- A name for the version
- Select month and year of the version of data

This will generate a template that will be downloaded on users’ end.

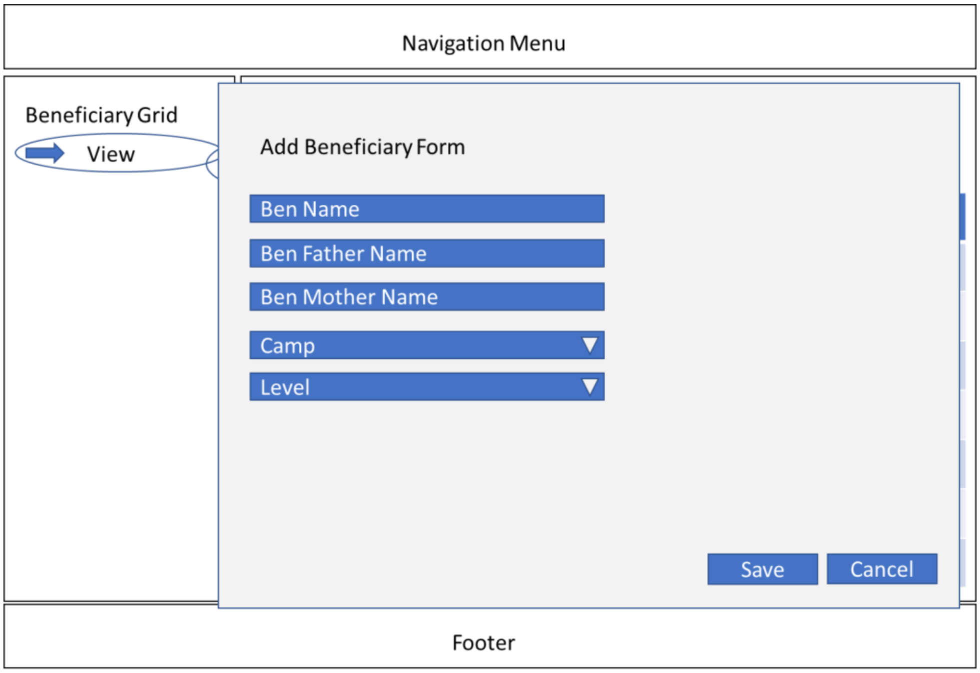
Users will fill up the template and then import the file into the system.

3.3.6. Edit data

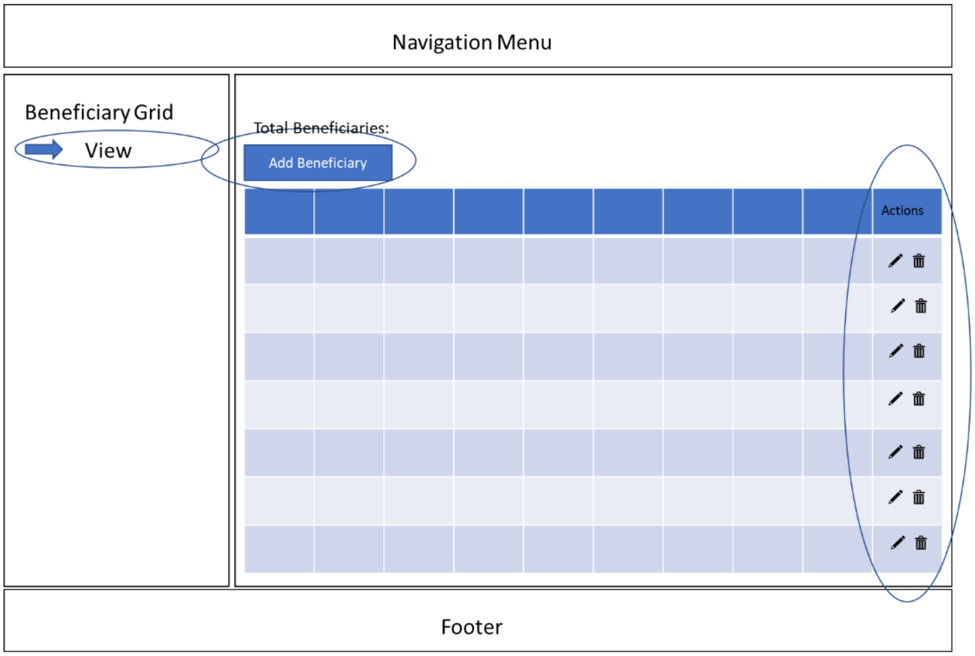


Users will be able to add new beneficiaries to the record.

Adding a new beneficiary individually will provide a form containing some information. (*UNICEF will provide the information that will be needed to add a new beneficiary*). Adding new beneficiary will only be allowed based on Fixed Column columns specified by the default view.



Users will fill out the form and add/save the data and system will store the data as a new row to represent a beneficiary data.



Based on permissions, users will also be able to edit individual beneficiary information. For every row, beneficiary data, there will be an action item that will allow users to make edit/delete, data of specific version (instance).

Editing a beneficiary data will provide a form. The form will contain all indicators of that beneficiary, these indicators can be searched. Their data will be prefilled based on version specific.

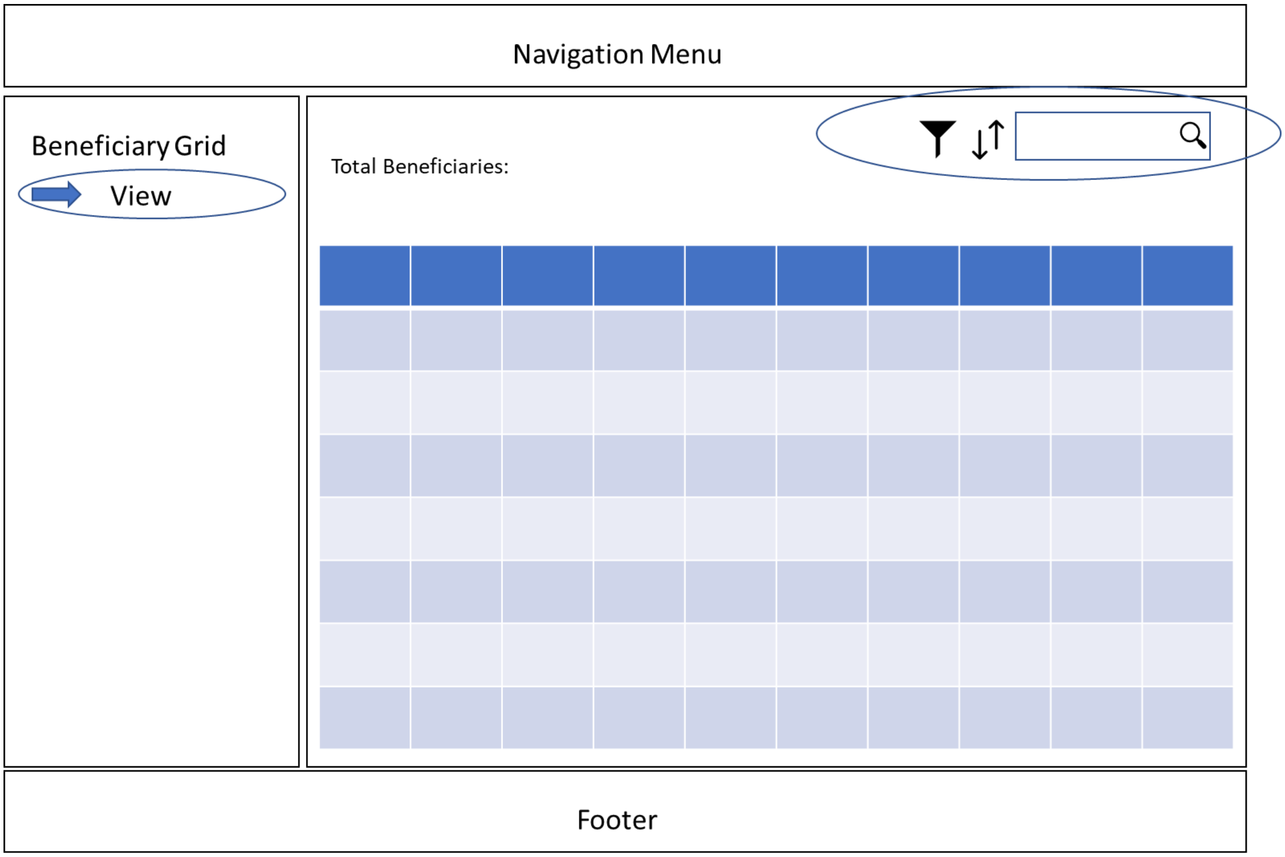
Deleting a beneficiary data will delete the row from the grid. Deletion of the data will not show in the next instance (version).

Before deletion, a confirmation to delete message must be displayed.

Data available in the last instance (version) will be provided in the next instance. For example, if a row is deleted in June version, will not show in July.

Admins can also change/update beneficiaries' facility. This will change the beneficiaries' information in Facility grid.

3.3.7. Filter, search, and sort



Search functionality should be available for users to search for a specific, or a list of, beneficiaries. Searching for beneficiary is based up defined columns. Searching will be based on reasonable number of columns. Also, must be from Fixed Columns. (*Search columns will be confirmed by UNICEF*).

Filter functionality should be available for users to filter beneficiaries by columns. Filter should be based on reasonable number of columns. Filter will be done on Fixed Columns. (*Filter columns will be confirmed by UNICEF*).

Sorting functionality should be available for users to sort beneficiaries according to ascending and descending order. (*Sorting columns will be confirmed by UNICEF based on Fixed Columns*)

3.3.8. Duplicate entries

From the first import, system will allow duplicate data to be imported.

From that existing data duplicates, system will generate a report (See section 3.8) for the duplicate entries.

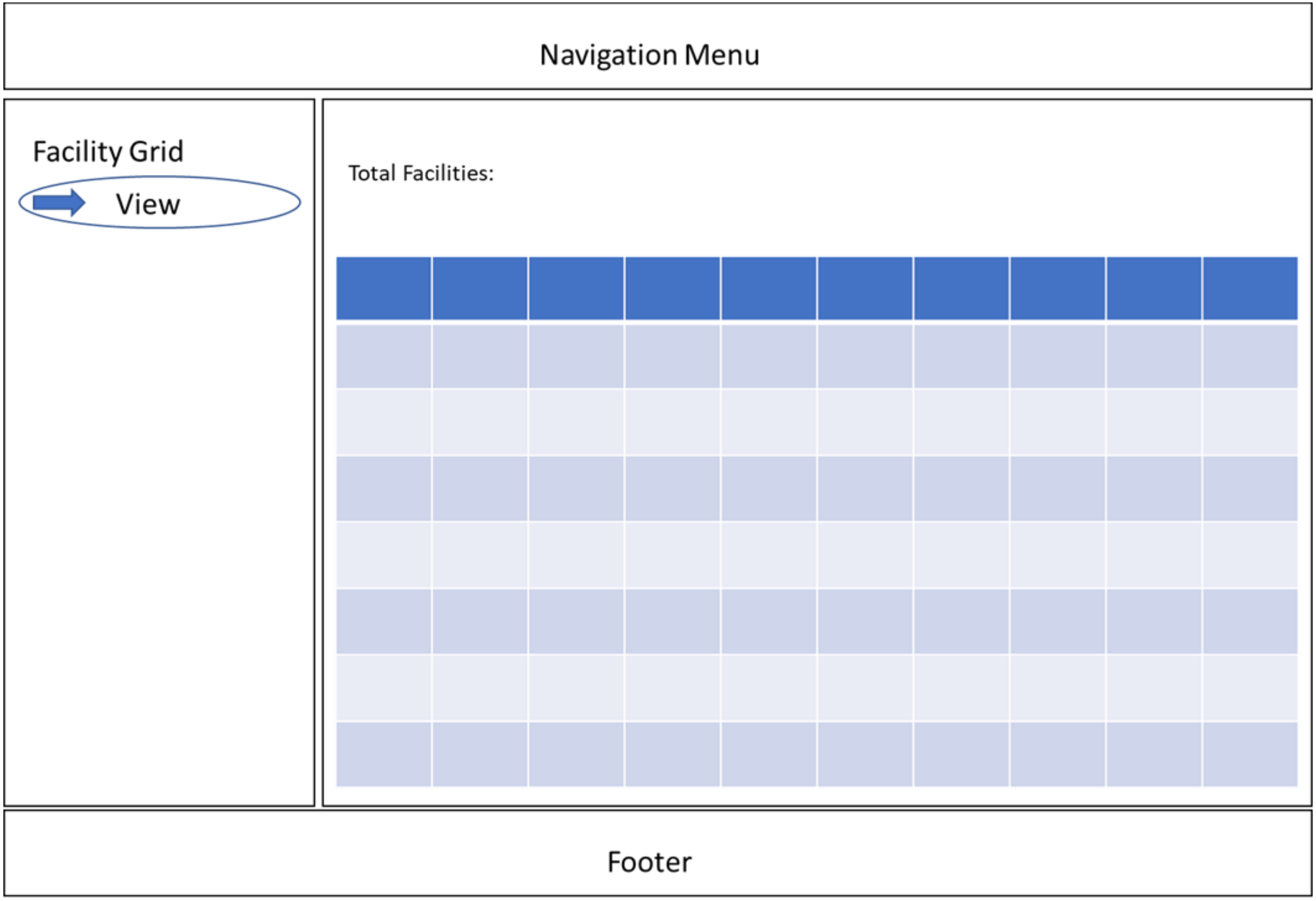
After this, system will not allow duplicates entries to be added either from imports. (***The duplicates will be prevented by the validation parameters provided by UNICEF***).

If users add a new beneficiary, system will use the columns such as "*Progresss_ID*", Name, Father's Name, etc. to check if the entry already exist in the grid. (*UNICEF will provide the parameter using which duplication will be searched*).

3.4. Facility Data

Learning Facilities, or Facilities, are similar places that help beneficiaries to get educations. These data are collected once and some of these data are updated over different periods. These data are needed to be shown in a grid view (table view) to represent all the columns of all the facilities that Education sector is working with.

3.4.1. Facility Data Grid



There will be a grid that is dedicated to show all the Facilities.

Each column will contain a property for a Facility.

When selected from the menu, all the Facilities' along with their default columns (*will be confirmed by UNICEF*) must be shown in columns, e.g.: Facility, Camp, Camp ID, Location, etc.

Each column will contain a type and based on those types; data will be stored under those columns. Different types of columns:

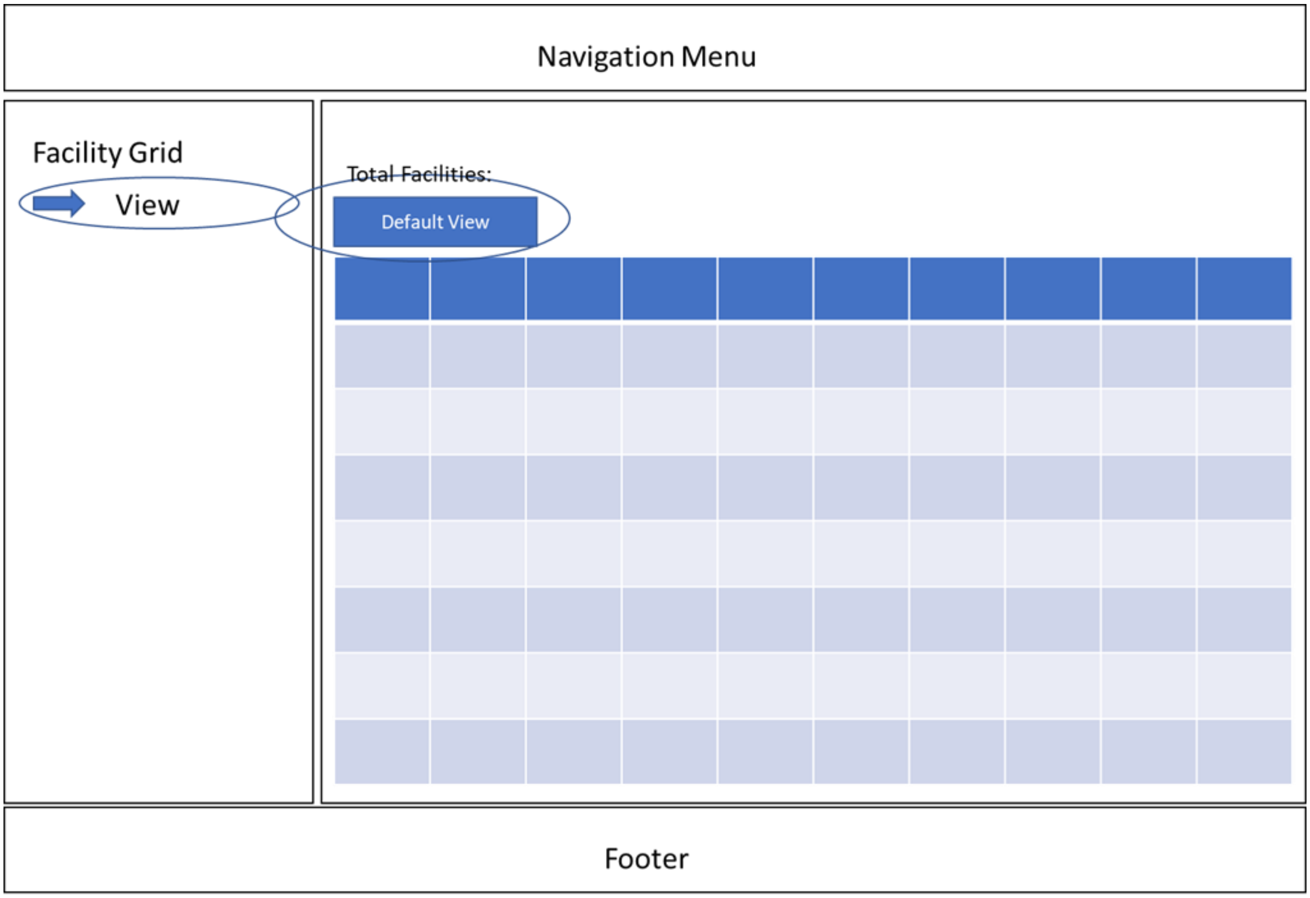
- Text
- Numeric
- Decimal
- Equation
- Date

- Boolean
- List

All data collected will be stored under this grid.

On the top, it will show the total number of Facilities in the grid view.

3.4.2. Customized Views for Grid Data



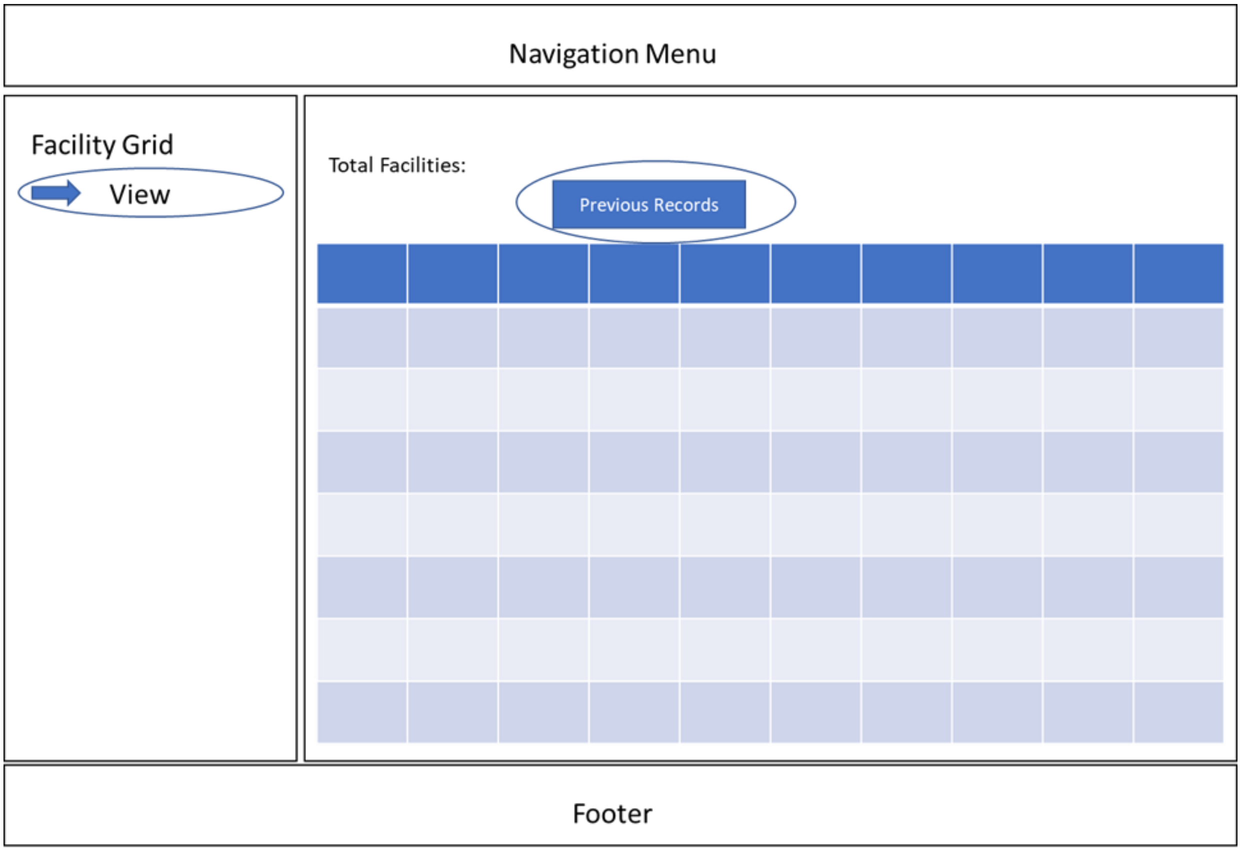
In default, the grid will not show all the Facilities' basic data, e.g.: Name, Father's Name, Mother's Name, etc. Default view will only show Fixed Columns. *[These basic columns will be defined by the UNICEF]*

Users can change the view according to their specific needs.

Users will be able to select the columns from a columns' list and create a new view for the Facilities.

Users will also be able to save and set a name for those views.

3.4.3. Store records in different dates



Data stored in the grid can be updated over time.

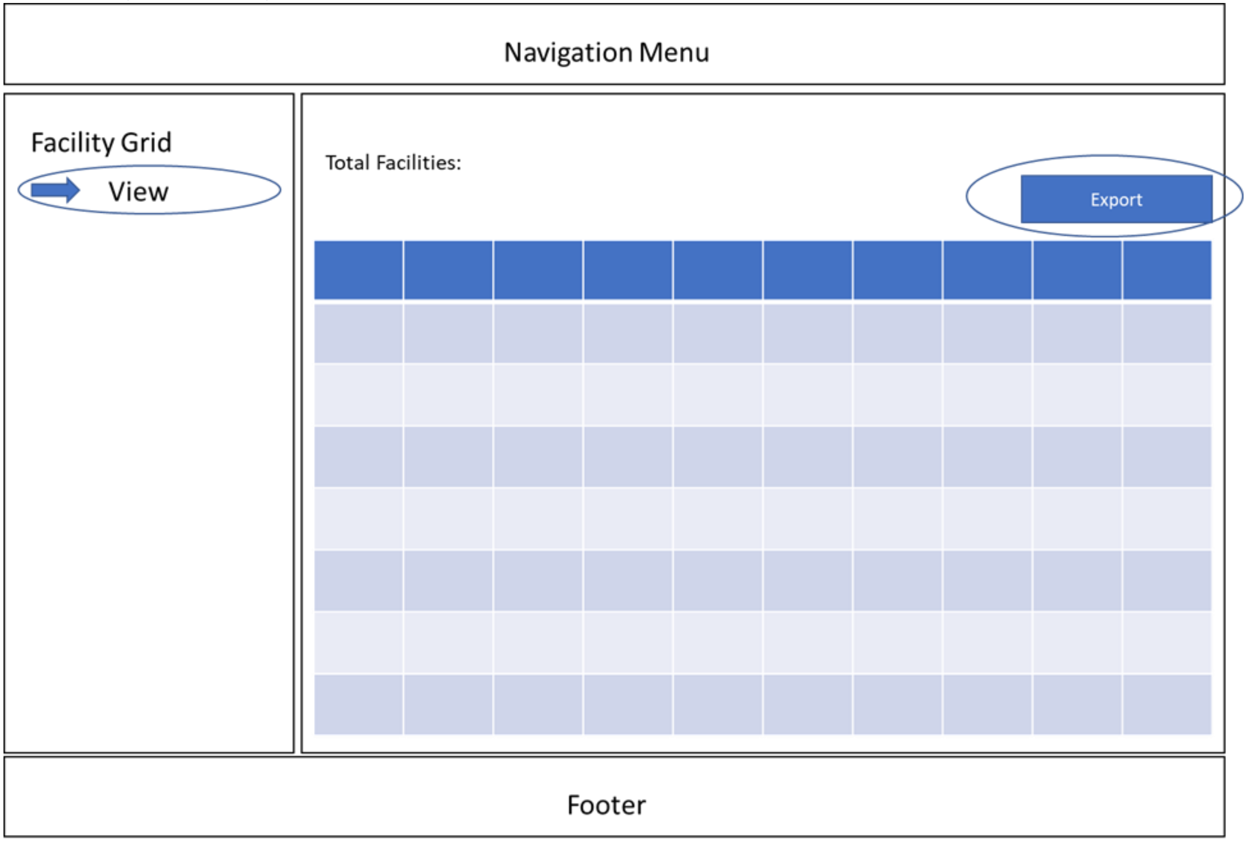
Records will be stored from different collection instances, in different time, using the monitoring section (see Section 3.5), users will be able to access those instances.

Users can choose a specific collection instance which will load the data collected from that point of time in the Facility grid.

The instances can be searched and then selected. Once the instance is selected the grid will show the data of that collection instance.

By default, in the Facility grid, it will always show the latest instance of data.

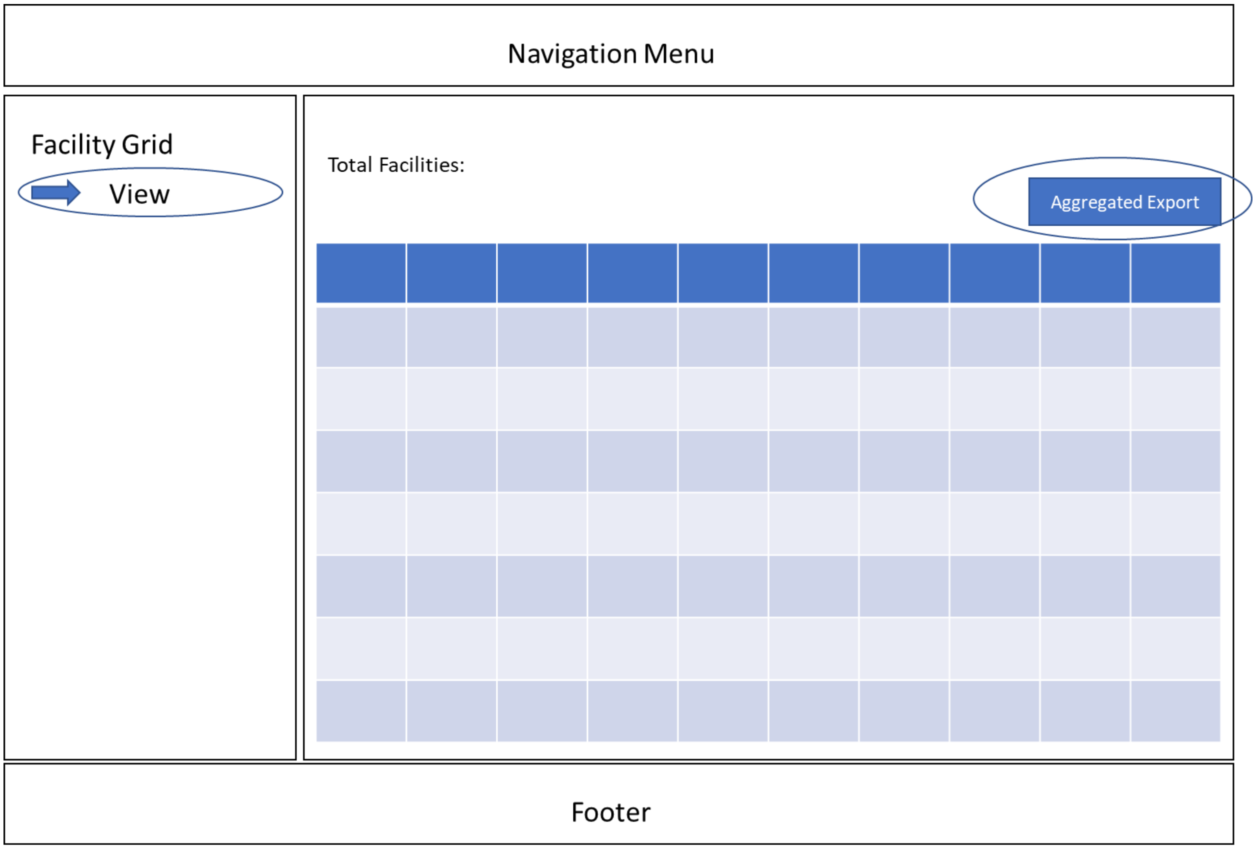
3.4.4. Export Facility Data



Users will be able to export the data from this grid.

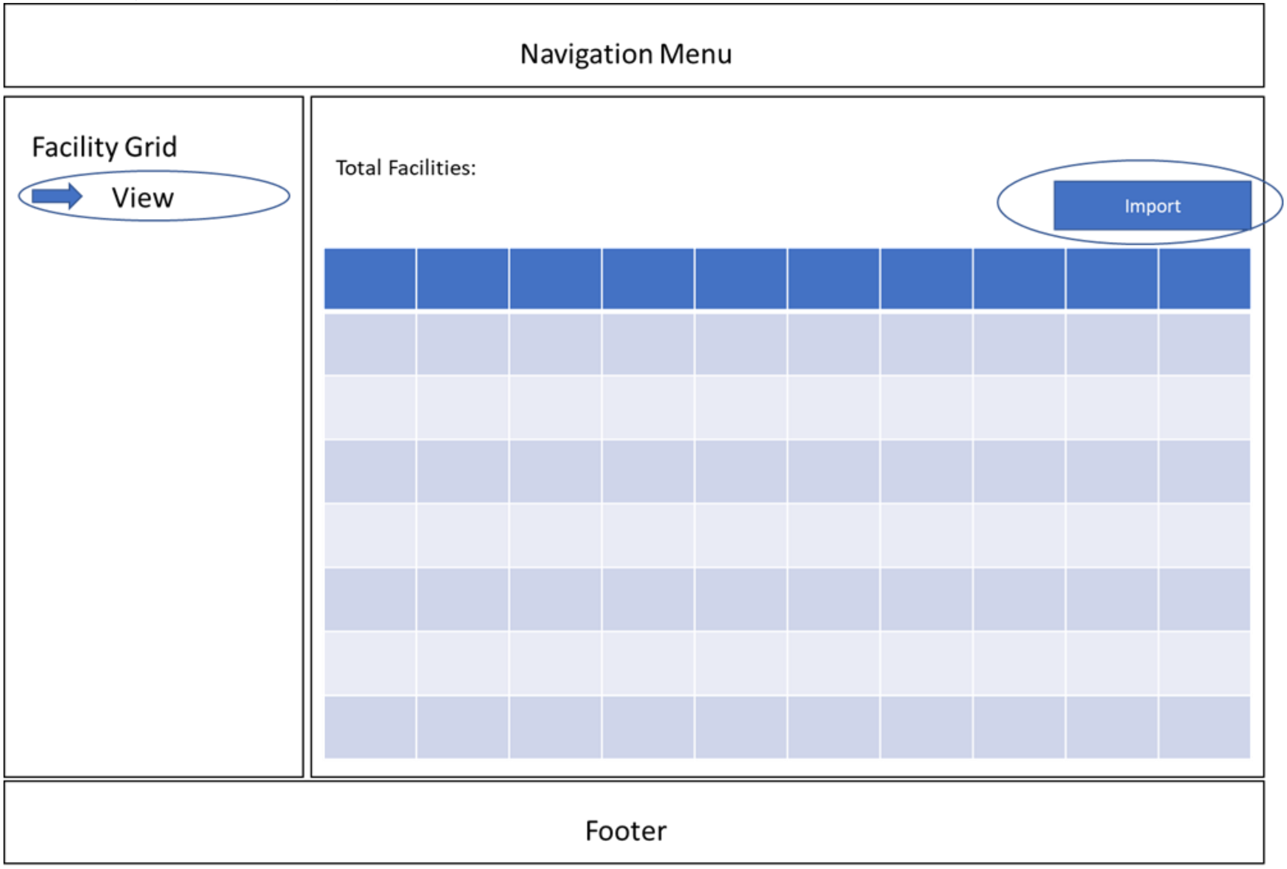
This will export all the columns from the Facility grid.

Export will offer excel, csv file format and allow to share on Google Drive.



There will also be an Aggregated export feature that will allow users to download all the data, Fixed Columns, dynamic columns, data from different instances, etc. will be generated into an Excel or CSV file or Google Drive share. [*The structure of this file will be provided by UNICEF*]

3.4.5. Import bulk Facility data



Users will be able to import Facilities and add them to the grid.

Import from an excel file.

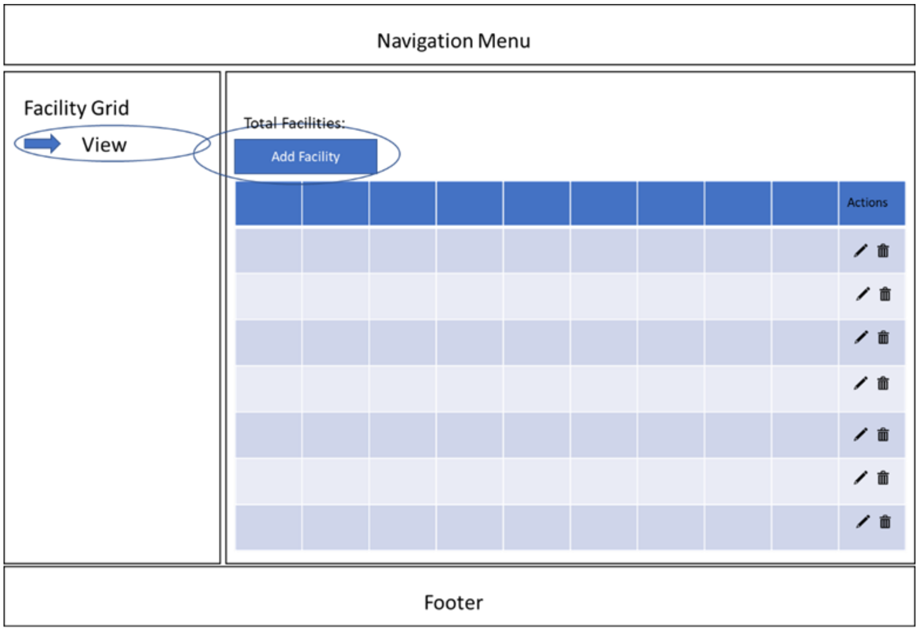
Based on the columns, users will have to create the dynamic columns first, in the grid. This also allow them to import data with dynamic columns.

Import will first download a template. The template will be used arranging the Facility data for import.

Import Facility data will include data validation to check for erroneous data content in the template file.

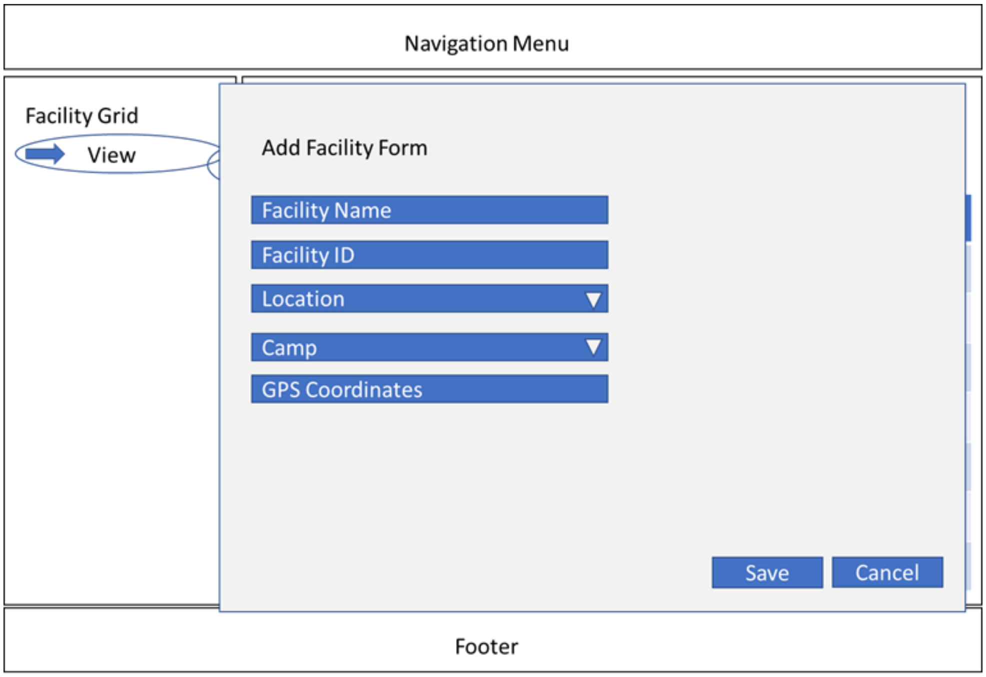
While importing, system will run data validation to check for erroneous data. If the system finds error in data while importing, it will show messages to identify where in import file is the error.

3.4.6. Edit data

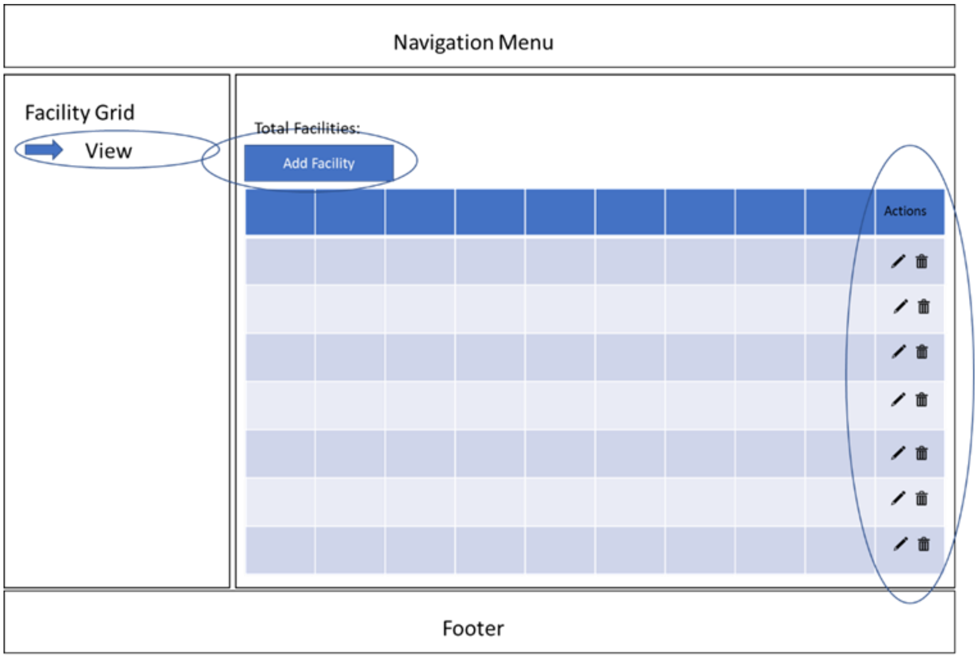


Users will be able to add new Facilities to the record.

Adding a new Facility individually will provide a form containing specific indicators (columns). (*UNICEF will provide the information that will be needed to add a new Facility*). Adding new Facility will only be allowed based on any columns that are not in the default view but is a Fixed Column.



Users will fill out the form and add/save the data and system will store the data as a new row to represent a Facility data.



Based on permissions, users will also be able to edit individual Facility information. For every row, Facility data, there will be an action item that will allow users to make edit/delete, the data of specific version (instance).

Edit and/or delete will be performed only on the selected instance.

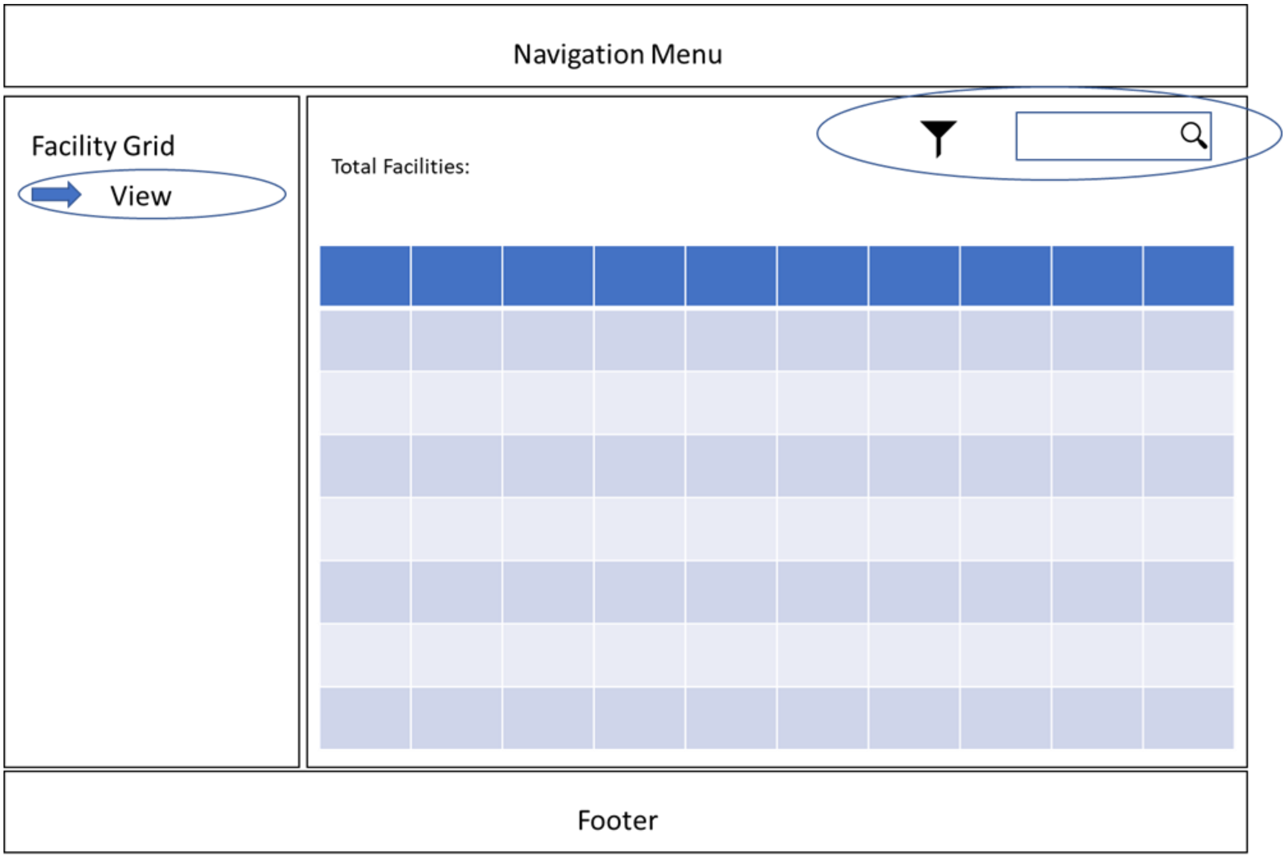
Editing a Facility data will provide a form. The form will contain all indicators of that Facility, these indicators can be searched. Their data will be prefilled based on version specific.

Deleting a Facility data will delete the row from the grid. Deletion of the data will not show in the next instance (version).

Before deletion, a confirmation to delete message must be displayed.

Data available in the last instance (version) will be provided in the next instance. For example, if a row is deleted in June version, will not show in July.

3.4.7. Filter, search, and sort

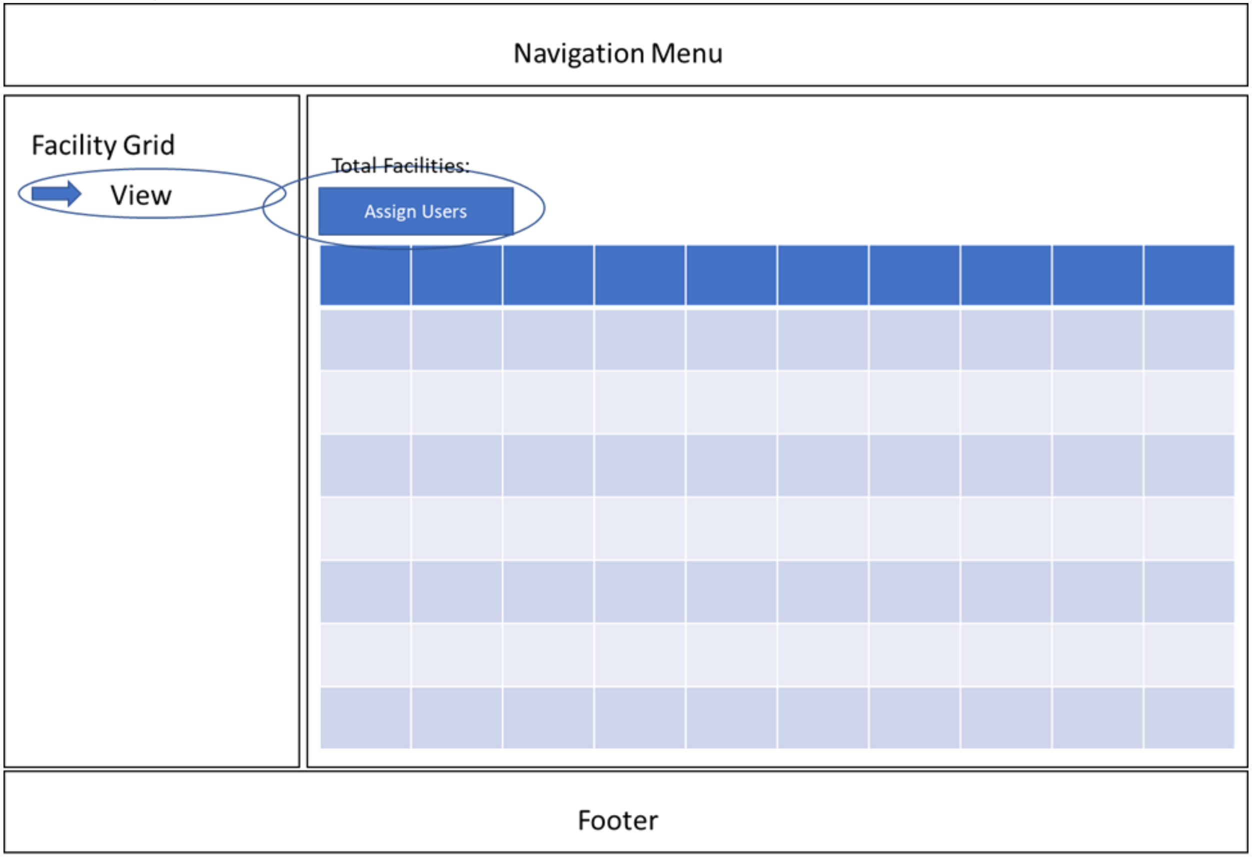


Search functionality should be available for users to search for a specific, or a list of, Facilities. Searching for Facility is based up defined columns. Searching will be based on reasonable number of columns. Also, must be from Fixed Columns. (*Search columns will be confirmed by UNICEF*).

Filter functionality should be available for users to filter Facilities by columns. Filter should be based on reasonable number of columns. Filter will be done on Fixed Columns and reasonable number of filters. (*Filter columns will be confirmed by UNICEF*).

Sorting functionality should be available for users to sort Facilities according to ascending and descending order. (*Sorting columns will be confirmed by UNICEF based on Fixed Columns*)

3.4.8. Assign users

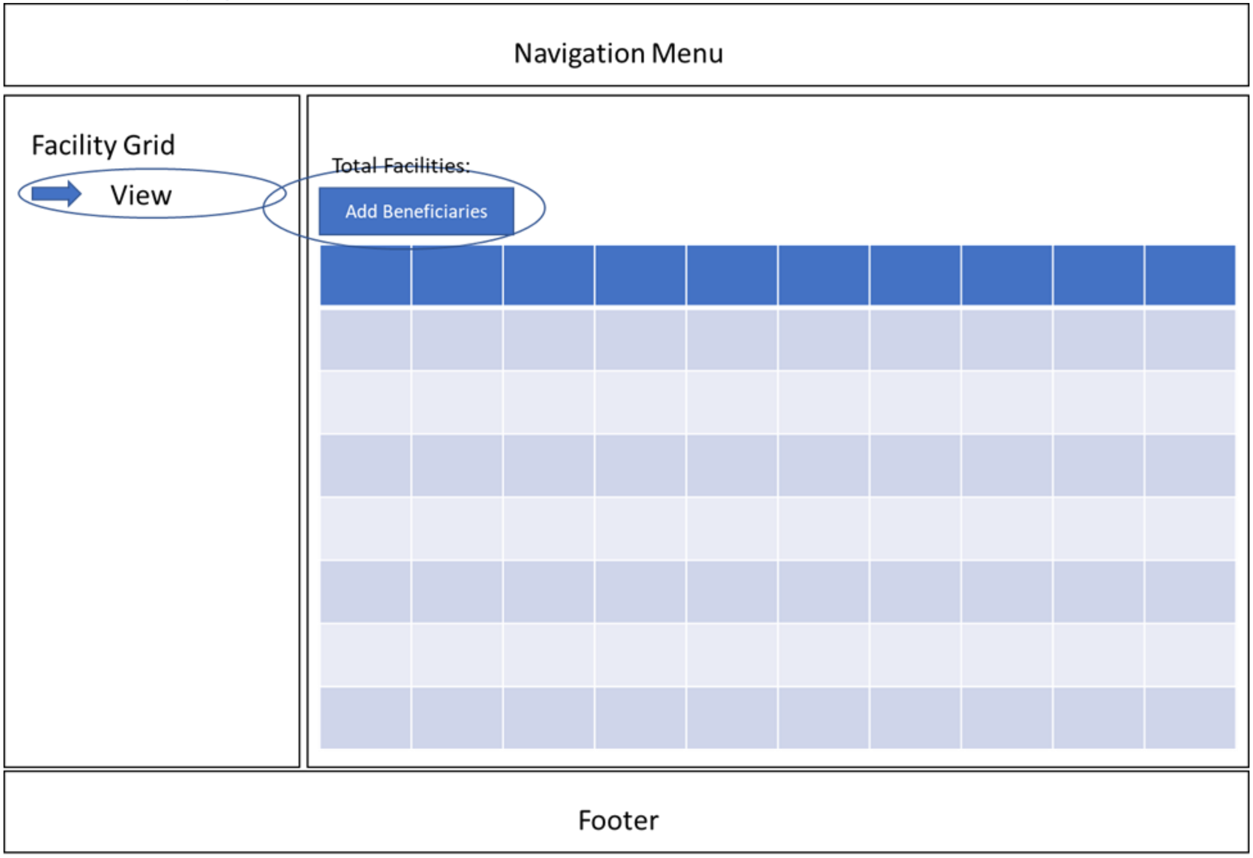


Data collection for beneficiaries and facilities are done by users (usually by Teacher level users).

Multiple facilities can be assigned to users with permissions.

Users will be able to assign Teacher level users to facilities. Teachers assigned to facilities, will be to view beneficiaries assigned to that specific facility.

3.4.9. Identifying beneficiaries



Each facility will be containing number of beneficiaries from the beneficiary grid.

Data collection process happens through the learning facilities.

Admins will be able to assign multiple beneficiaries, from beneficiaries' grid data, to a facility. (Initially when the data is imported, beneficiaries that already have Facility ID on their property will be assigned to the respected Facility).

Beneficiaries assigned to a facility will also be shown in beneficiary grid in their respected columns.

Admins will select the facility and then add beneficiaries to their designated facilities.

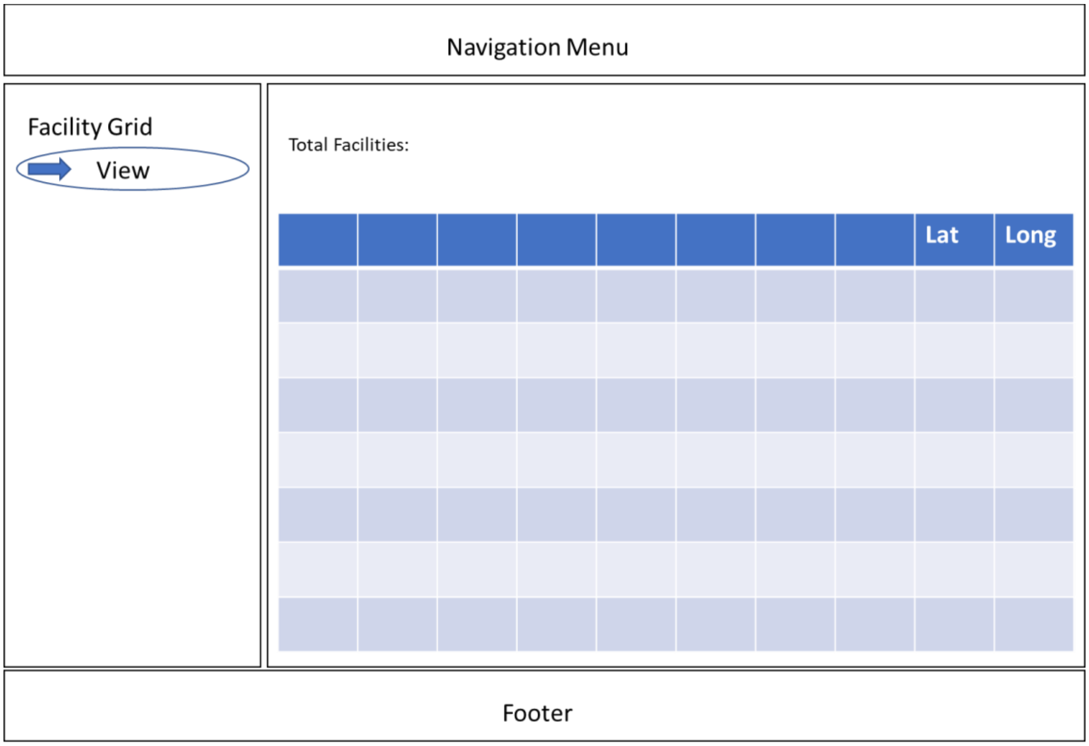
Admins can also change/update beneficiaries' facility.

While adding beneficiaries, Admins will be able to filter beneficiaries according to their columns.

Beneficiaries' information under each facility will be used to generate calculations in the 5W reports (See section 3.8) and Dashboard (See section 3.9). [*The fields and Arithmetic operations required should be provided by UNICEF before development starts*].

3.4.10. Geo Location

Every facility will have two columns: Latitude and Longitude. The combination of these two columns represents the geological location of the facility. These latitude and longitude information will be linked to each of the facilities.



For each facility that will be added will contain these two coordinates columns input. Geo coordinates will be stored as decimal data.

Geo Location Data Validation

Geo locations for the facilities will be validated based on the "*Shapefiles*" that are provide by UNICEF.

Shapefiles are format for storing the geometric location and attribute information of geographic features. Geographic features in a shapefile can be represent by points, lines, or polygons (area or boundaries).

Once the shapefiles are provided by UNICEF, they will be stored in the system for validating the geo locations for facilities against the geo locations columns, Latitude and Longitude.

Data provided for Geo Location columns, for the facilities, need to be cross checked against the shapefile data that is input in the system.

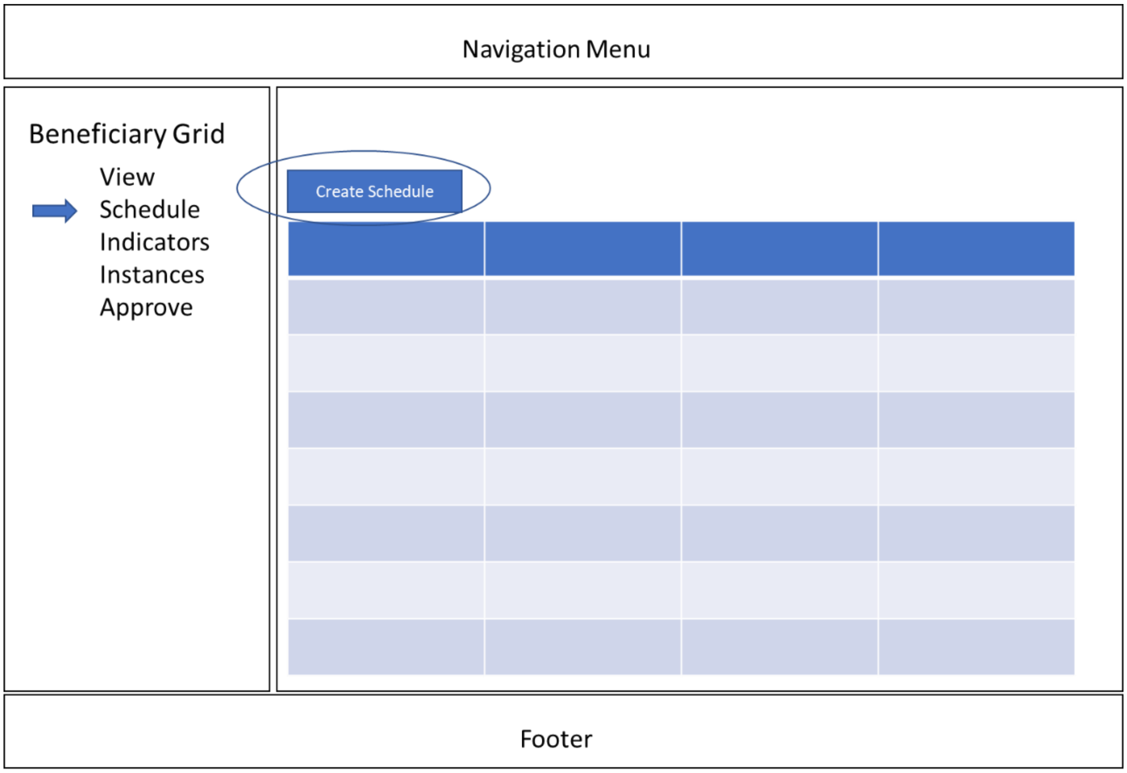
3.5. Monitoring

From time to time, users will conduct surveys using a form, answers of which will be collected and stored against the beneficiaries and facilities columns, to monitor the progress of the Program. Users will be collecting these data over a period and the system will be able to automate the process.

Monitoring will be done separated for Beneficiaries and Facilities.

3.5.1. Beneficiary Monitoring

For beneficiary data collection (monitoring), beneficiaries will be assigned to facilities. Facilities will assign teachers for data collection. Teachers assigned to specific facilities will get the beneficiaries assigned to them.



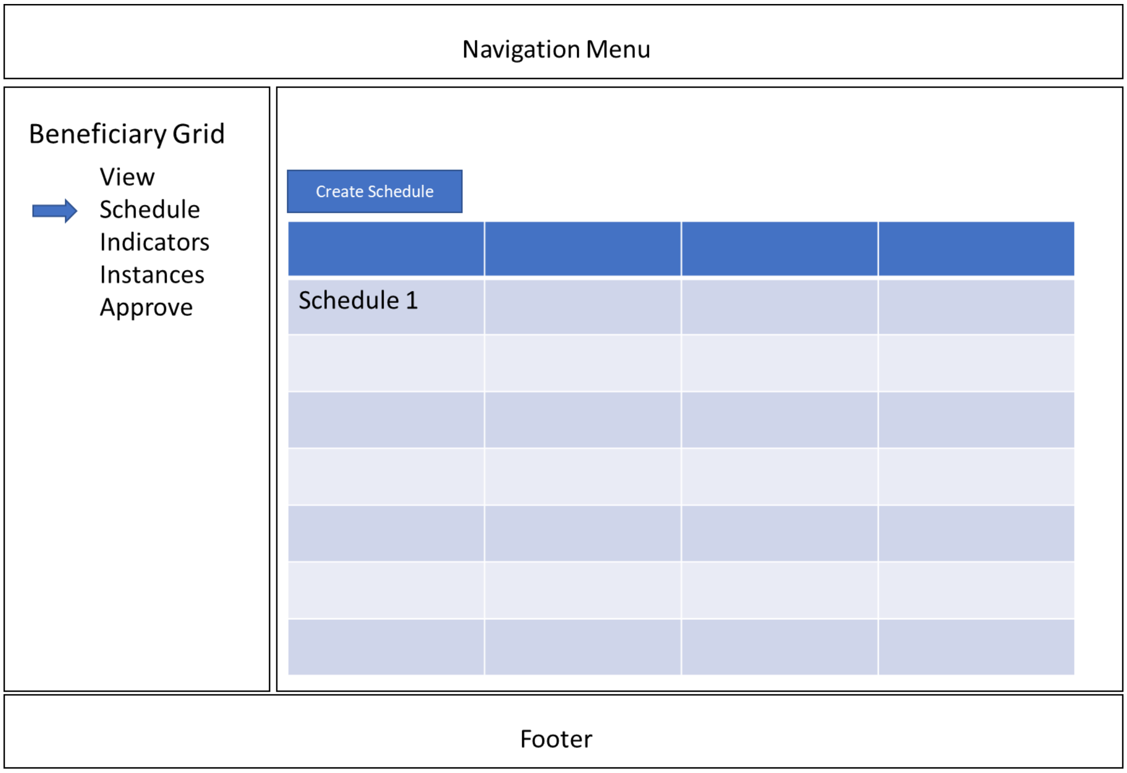
Data collection can be performed for all beneficiaries. Teachers assigned to facilities, will be able to view the data of all the beneficiaries assigned to that facility. Assigned teachers can view the updated data for all the beneficiaries assigned to that facility and choose to update selectively or for all the beneficiaries.

For example: Teacher A is assigned to Facility 21. Facility 21 is assigned with 100 beneficiaries. When Teacher A uses the application, he will be able to view all 100 beneficiary's data. He can choose to update the data for 5 beneficiaries or all 100 of them.

Users will be able to create a schedule for data collection process.

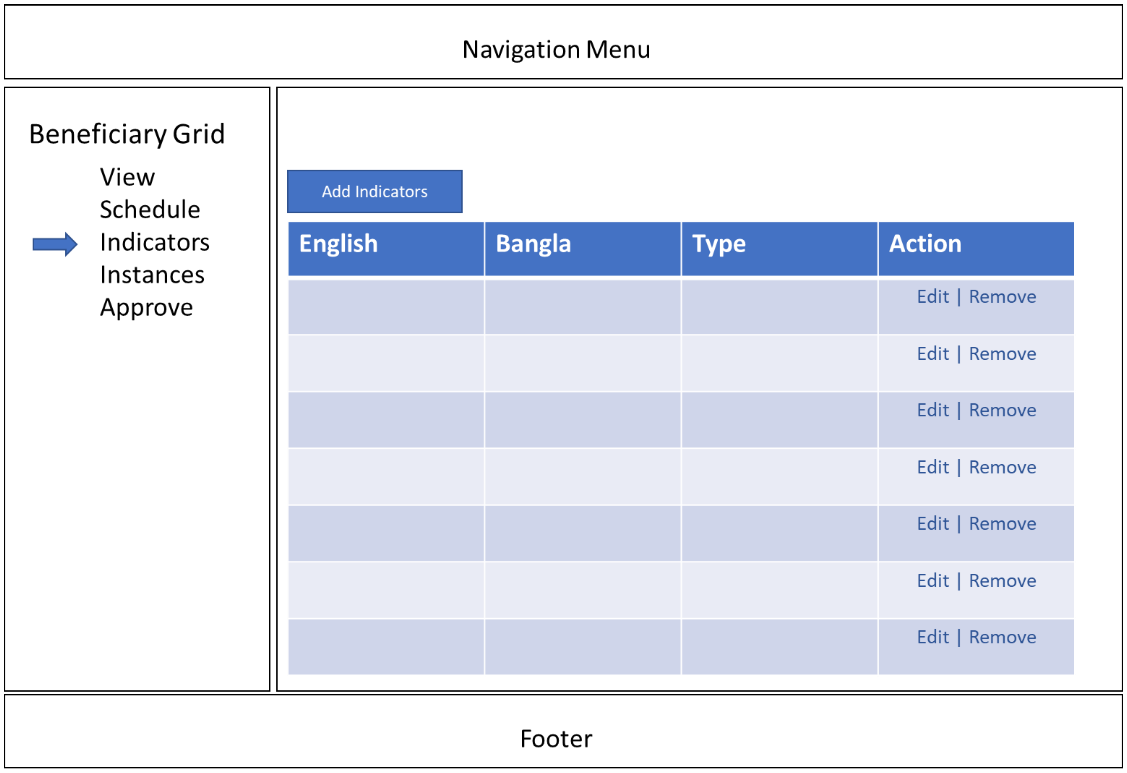
To create a new schedule, user will have to provide:

- Name for the schedule
- Frequency (Monthly/Bi-weekly/Weekly)
- Start Date
- End Date



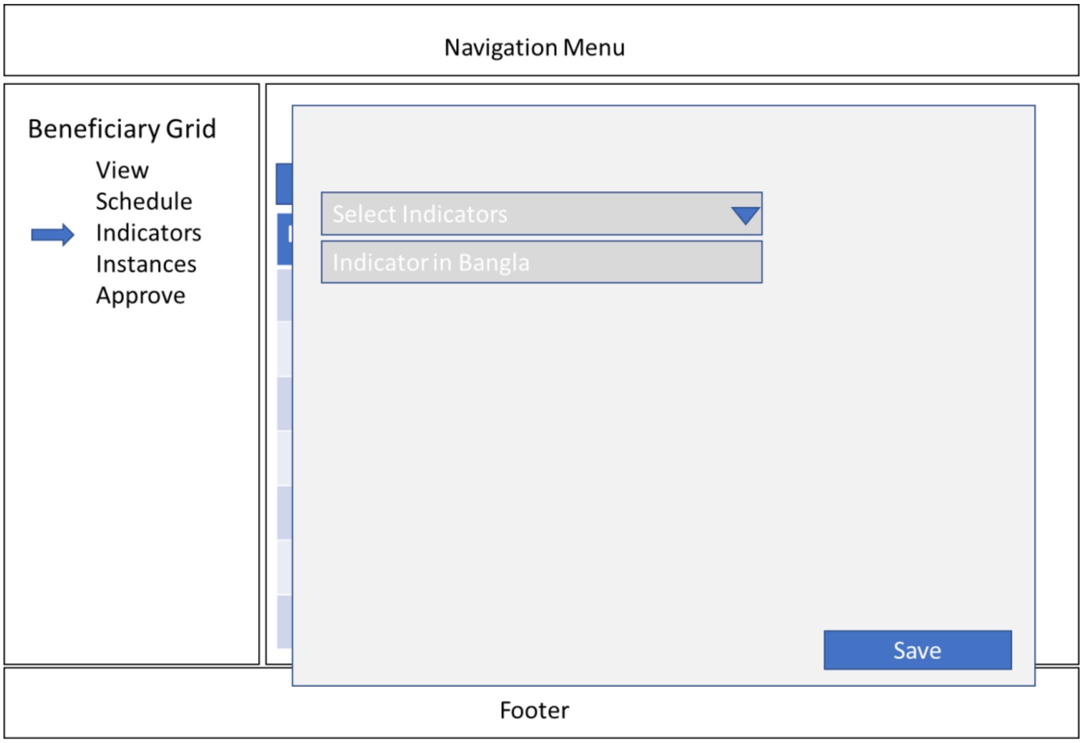
Schedule 1			

This will create a schedule. User will be able to delete schedule. There can only be one schedule at a time.



English	Bangla	Type	Action
			Edit | Remove
			Edit | Remove
			Edit | Remove
			Edit | Remove
			Edit | Remove
			Edit | Remove
			Edit | Remove

After a schedule is set, user will need to provide the list of indicators for which teachers will need to collect data with.



Users will be able to add indicators to the form from the list of columns from the beneficiary grid. Indicators will be the fixed and dynamic columns from beneficiary grid.

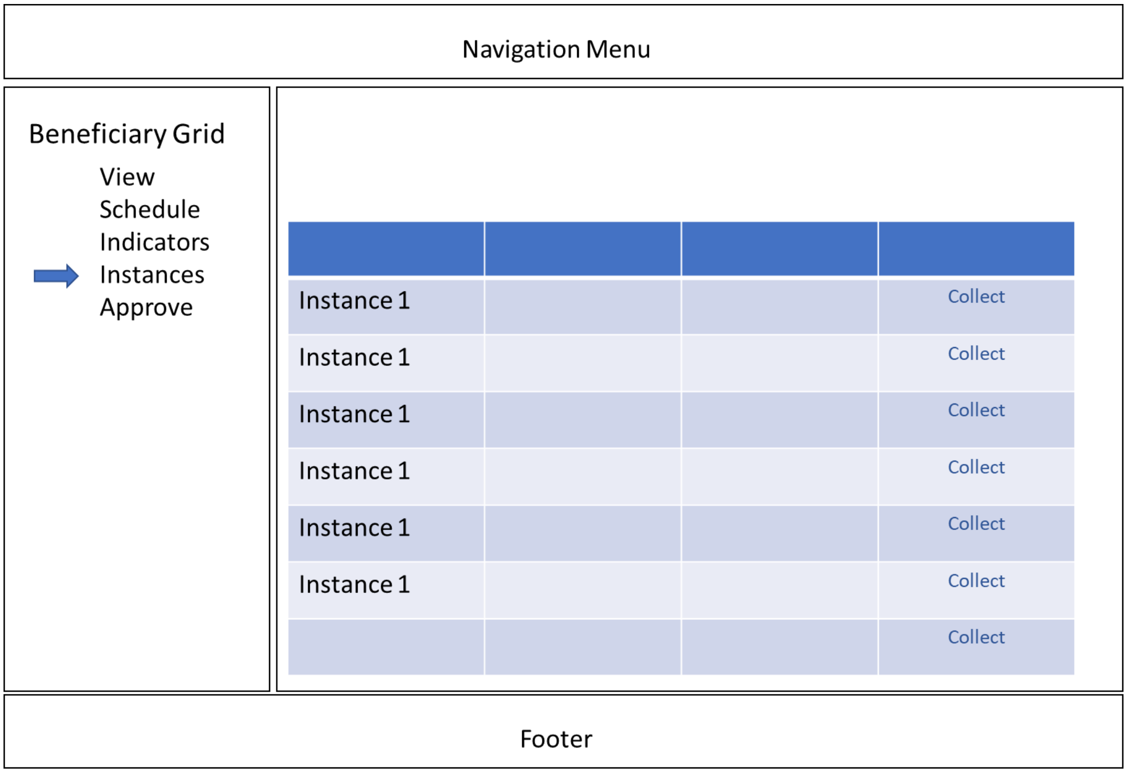
Users will be able to select an indicator and will be able to add a Bangla title to the indicator that will be shown on the data collection form.

Users can remove the indicators from data collection list.

Users will be able to re-arrange the indicators according to their needs.

The form will be shown on the teacher's device end for data collection (See section 3.6).

Based on the schedule the system will generate the number instances using the frequency and duration.



Navigation Menu

Beneficiary Grid

View
Schedule
Indicators
→ Instances
Approve

Instance 1			Collect
Instance 1			Collect
Instance 1			Collect
Instance 1			Collect
Instance 1			Collect
Instance 1			Collect
			Collect

Footer

Based on the schedule, Instances will be created.

In the instances section, user will find all the instances.

User cannot delete any instance.

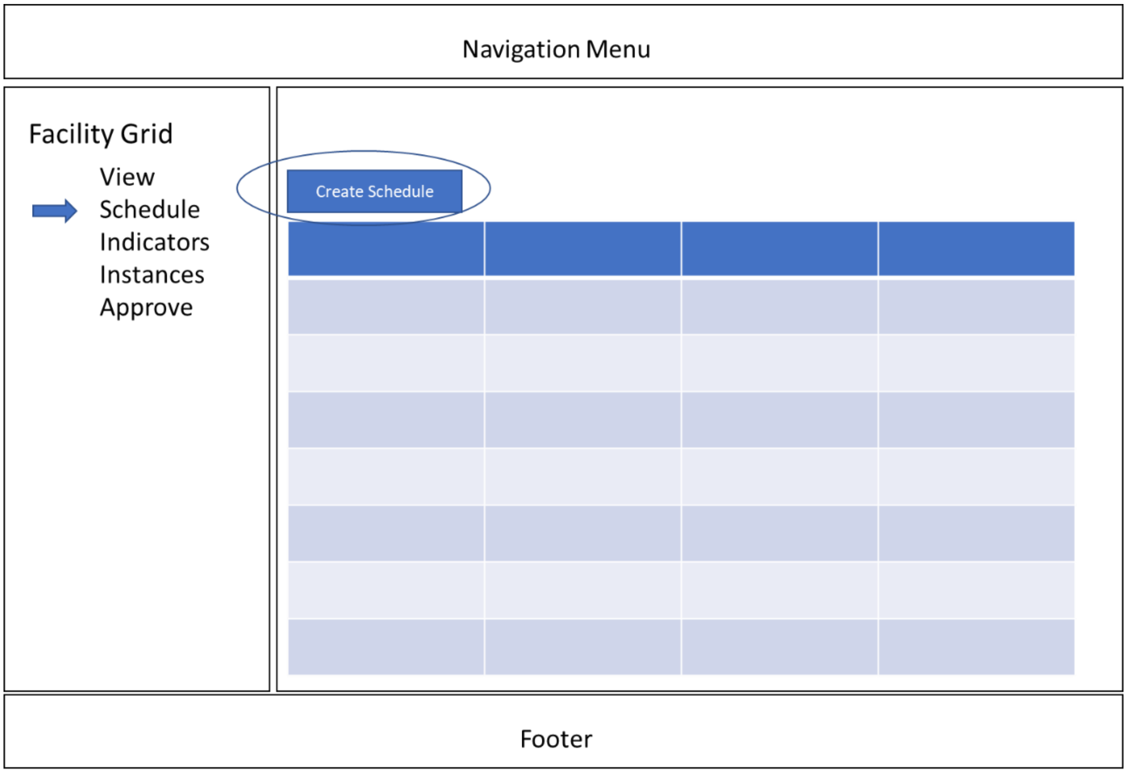
User can instruct an instance to start collecting data. Alternatively, the system will start the data collection instance automatically.

When the last record is approved, the system will automatically put the record collection instance to completed.

When selected for data collection, system will notify the assigned teachers for a collection run.

3.5.2. Facility Monitoring

From the grid, user will be able to select the number of Facilities and assign a teacher for collecting their data. Teachers will view the beneficiaries that are assigned to the facilities that they are assigned.



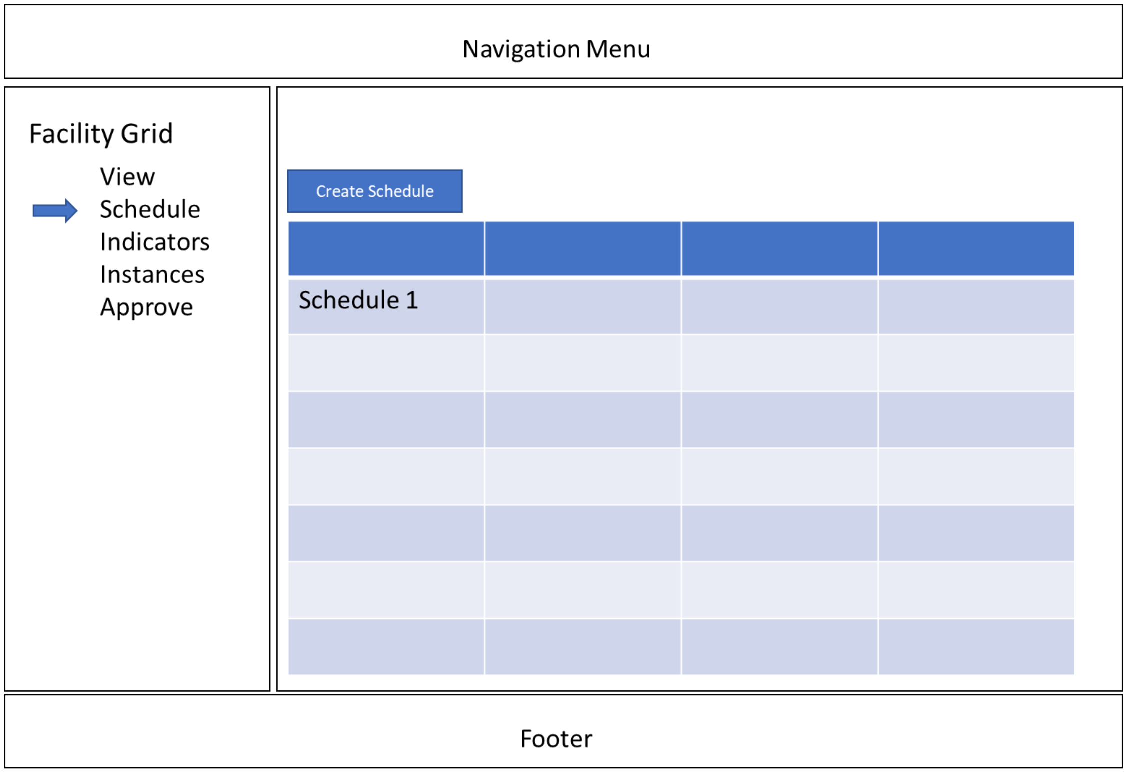
Assigned teachers can view the updated data for that facility and choose to update selectively or all the facility's data.

For example: Teacher A is assigned to Facility 21. When Teacher A uses the application, he will be able to view all the properties for Facility 21. He can choose update the data for 5 properties or all of them.

Users will be able to create a schedule for data collection process.

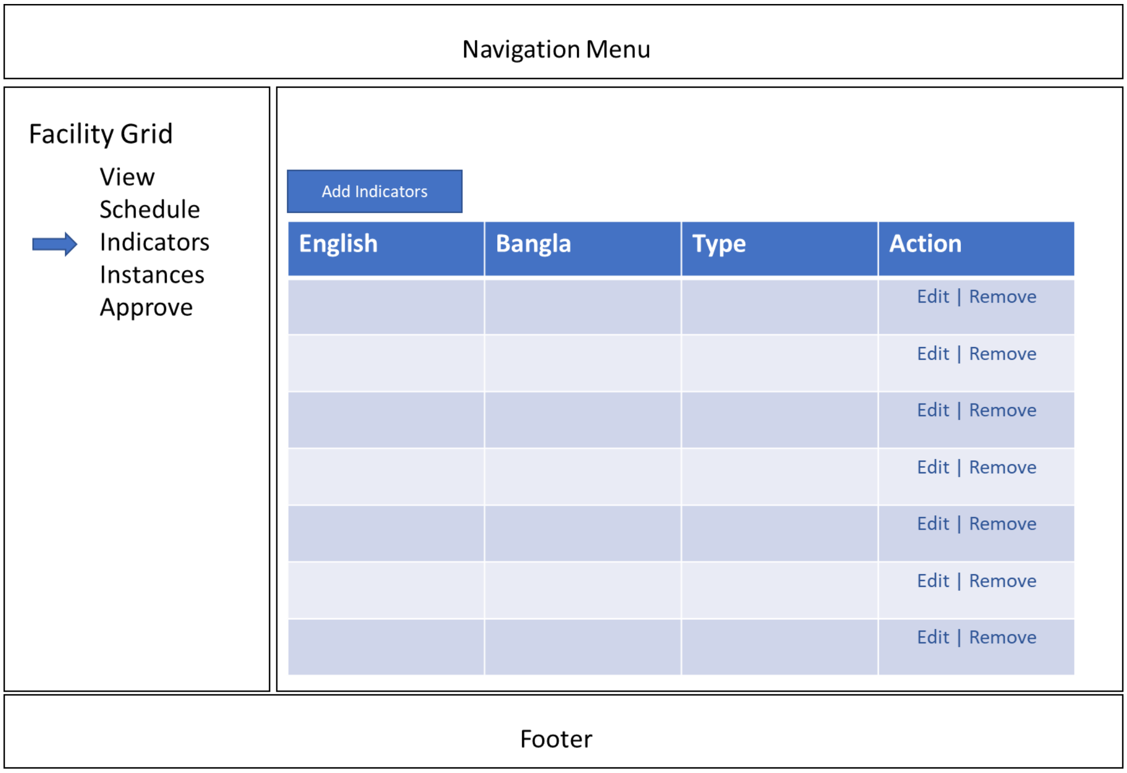
To create a new schedule, user will have to provide:

- Name for the schedule
- Frequency (Monthly/Bi-weekly/Weekly)
- Start Date
- End Date



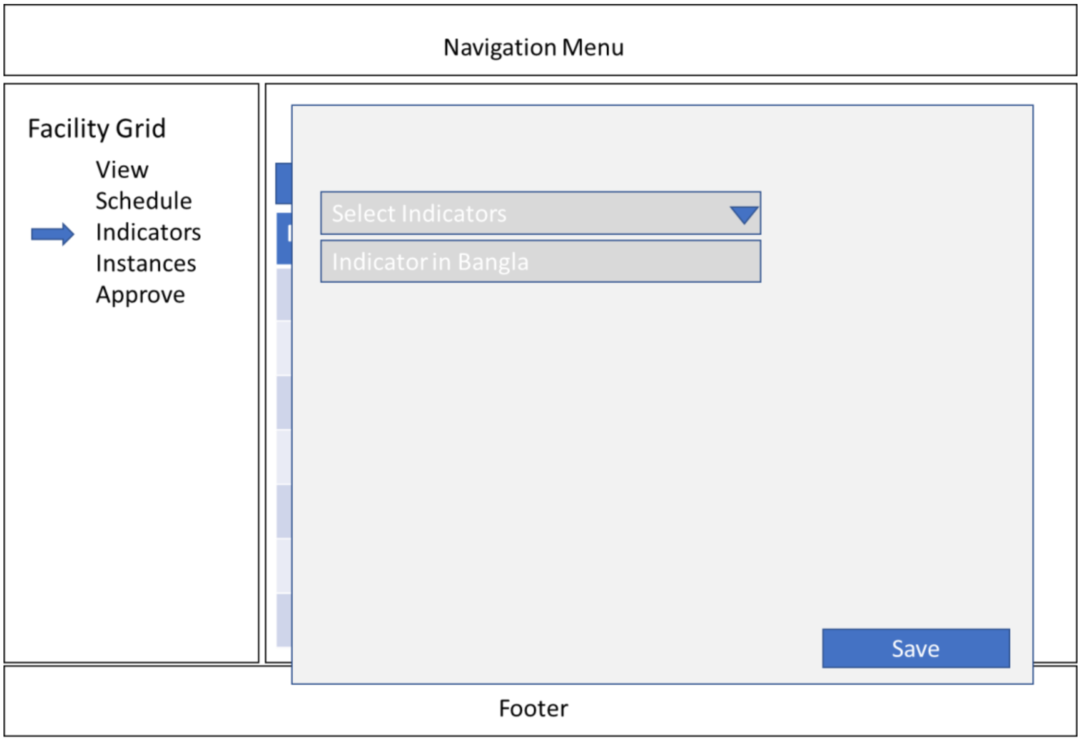
Schedule 1			

This will create a schedule. User will be able to delete schedule. There can only be one schedule at a time.



English	Bangla	Type	Action
			Edit | Remove
			Edit | Remove
			Edit | Remove
			Edit | Remove
			Edit | Remove
			Edit | Remove
			Edit | Remove

After a schedule is set, user will need to provide the list of indicators for which teachers will need to collect data with.



Users will be able to add indicators to the form from the list of columns from the facility grid. Indicators will be the fixed and dynamic columns from facility grid.

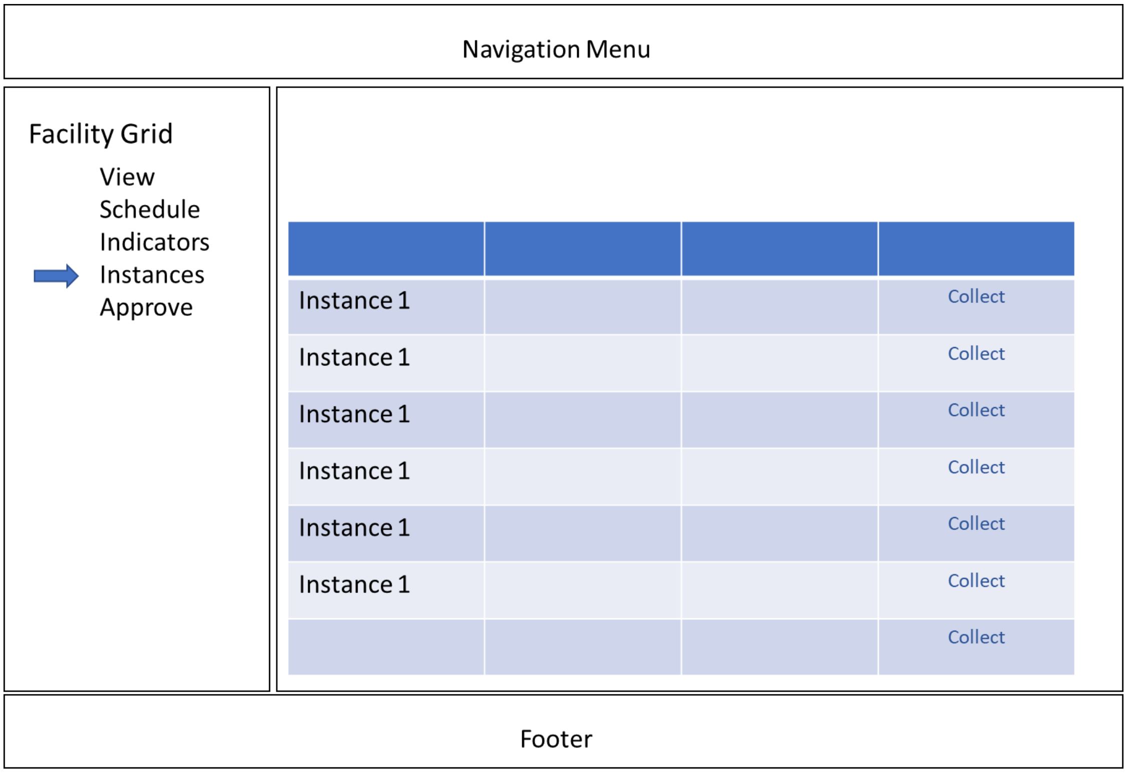
Users will be able to select an indicator and will be able to add a Bangla title to the indicator that will be shown on the data collection form.

Users can remove the indicators from data collection list.

Users will be able to re-arrange the indicators according to their needs.

The form will be shown on the teacher’s device end for data collection (See section 3.6).

Based on the schedule the system will generate the number instances using the frequency and duration.



Instance 1			Collect
Instance 1			Collect
Instance 1			Collect
Instance 1			Collect
Instance 1			Collect
Instance 1			Collect
			Collect

Based on the schedule, Instances will be created.

In the instances section, user will find all the instances.

User cannot delete any instance.

User can instruct an instance to start collecting data. Alternatively, the system will start the data collection instance automatically.

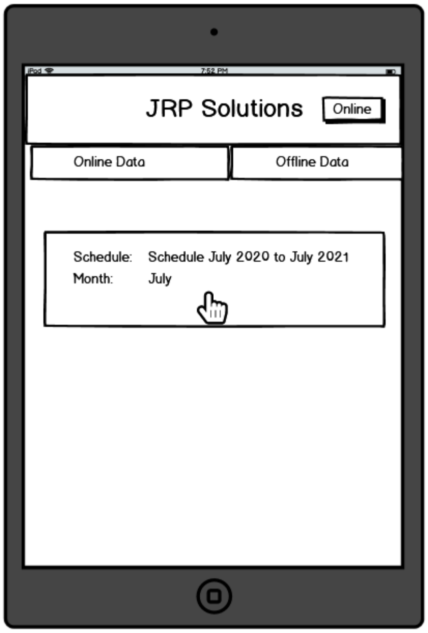
When the last record is approved, the system will automatically put the record collection instance to completed.

When selected for data collection, system will notify the assigned teachers for a collection run.

3.5.3. Data collection process

After monitoring for beneficiary and facility columns are set and teachers are assigned.

Teachers will be using Android based Tablet devices for collecting the data from the beneficiaries.

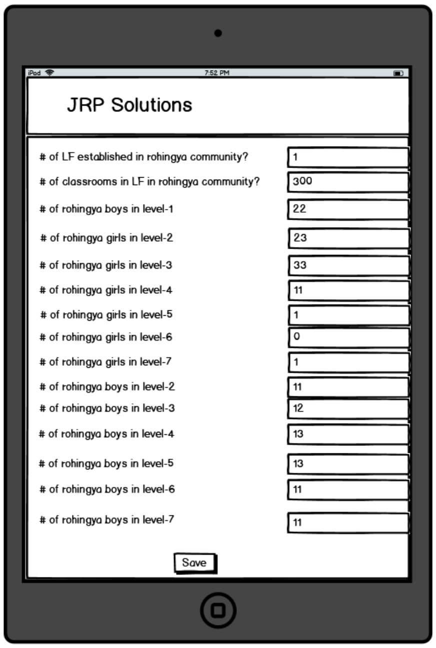


Schedule: Schedule July 2020 to July 2021
Month: July

On the device it will show the scheduled plan and offer teachers to collect data for offline or online. Offline data collection will store the collected data on the device and later sync with server side, for approval process, once connected to the internet. Online data collection will post the data for approval.

Teachers will login into the device and they will find latest created schedule on their device.

Teachers will be shown to collect data for beneficiaries and/or facilities if they are assigned to collect for both.

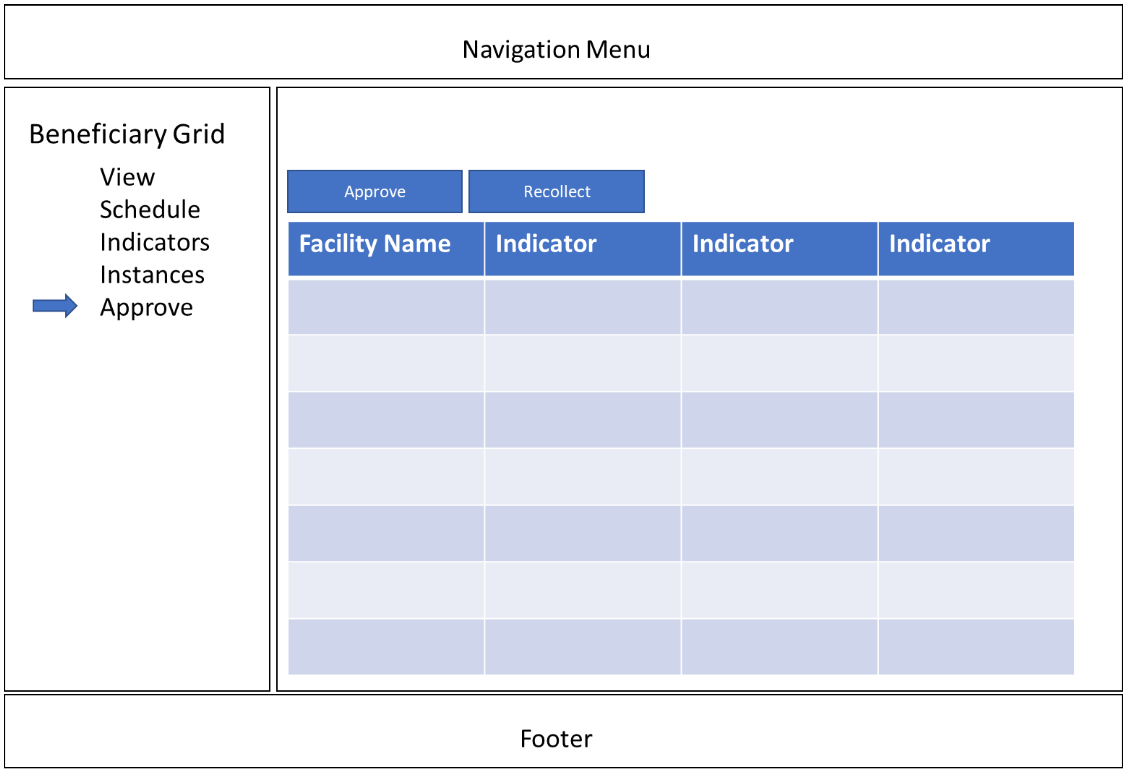


# of LF established in rohingya community?	1
# of classrooms in LF in rohingya community?	300
# of rohingya boys in level-1	22
# of rohingya girls in level-2	23
# of rohingya girls in level-3	33
# of rohingya girls in level-4	11
# of rohingya girls in level-5	1
# of rohingya girls in level-6	0
# of rohingya girls in level-7	1
# of rohingya boys in level-2	11
# of rohingya boys in level-3	12
# of rohingya boys in level-4	13
# of rohingya boys in level-5	13
# of rohingya boys in level-6	11
# of rohingya boys in level-7	11

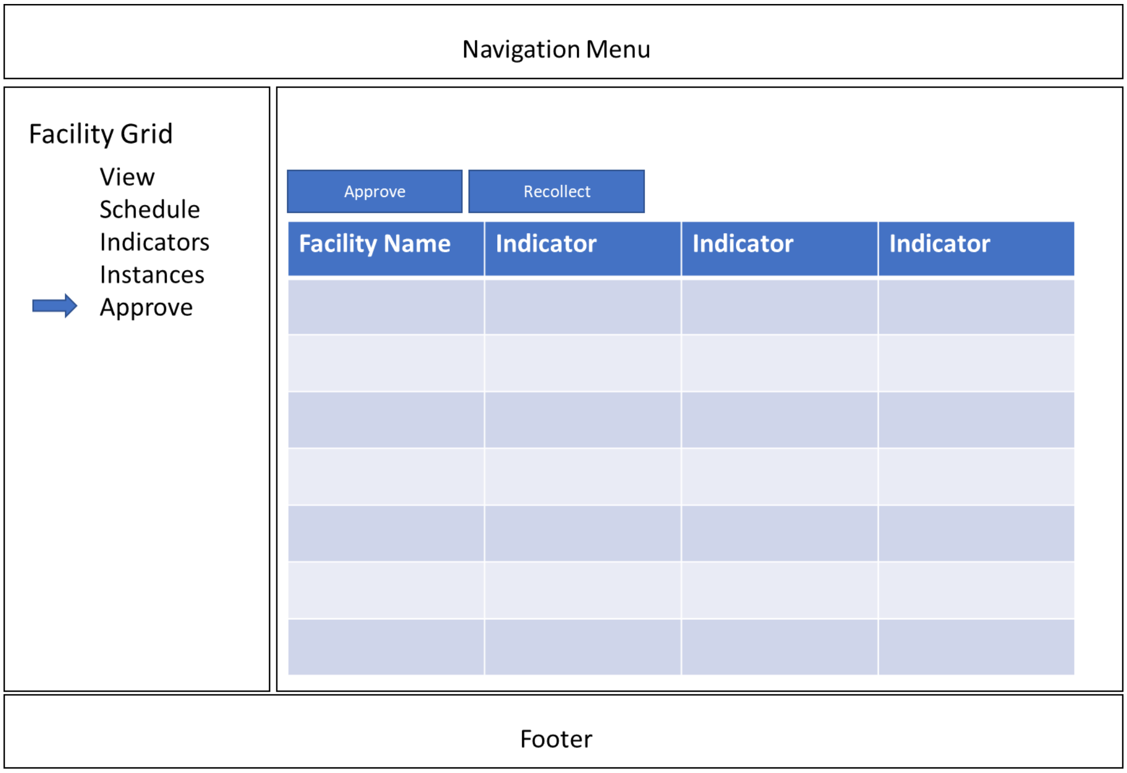
After selecting that, teachers will be shown the list of indicators for which they need to collect data for. This will be pre-filled with last collected data. After collecting the data, it will be reviewed for approval, inserted in the beneficiary and facility grids, or recollect for again notifying the teachers for to collect.

3.5.4. Data review

For both beneficiary and facility grid data collection process, supervisors of the collected data will be presented for review for approval or recollect.



Facility Name	Indicator	Indicator	Indicator



Facility Name	Indicator	Indicator	Indicator

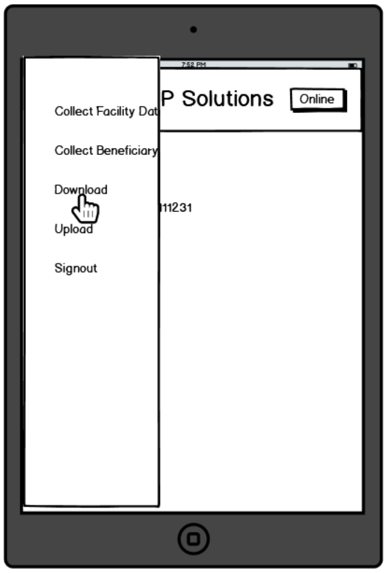
The collected data will be shown under the respected grid sections. Users will able to review the date and selecting them for actions.

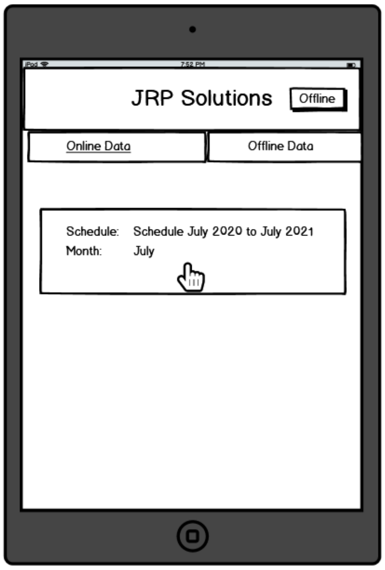
Once the data are selected, users can approve them and the data will be updated against the indicators of the beneficiary grid and of the facility grid, respectively.

Once the data are selected, users want to recollect them, a notification will be sent to the assigned teachers for recollecting those indicator data.

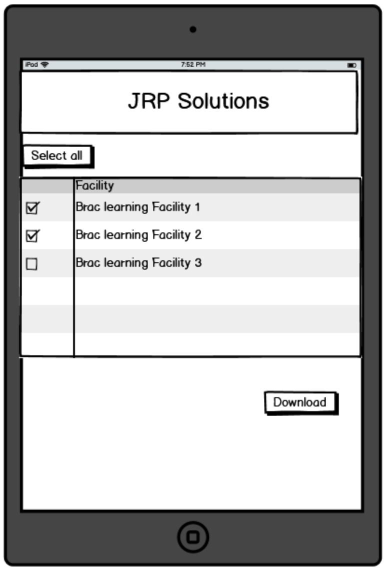
3.6. Mobile Application

Data collection process will be run using an Android based application that will allow users to collect data design based on the monitoring schedule and instances. The mobile device user will be notified of an instance when it is the time run the instance.





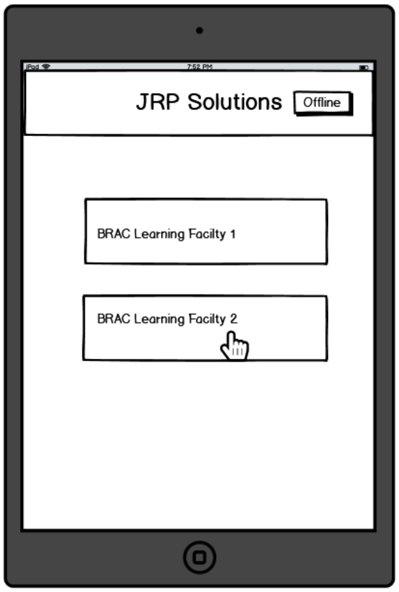
Schedule: Schedule July 2020 to July 2021
Month: July

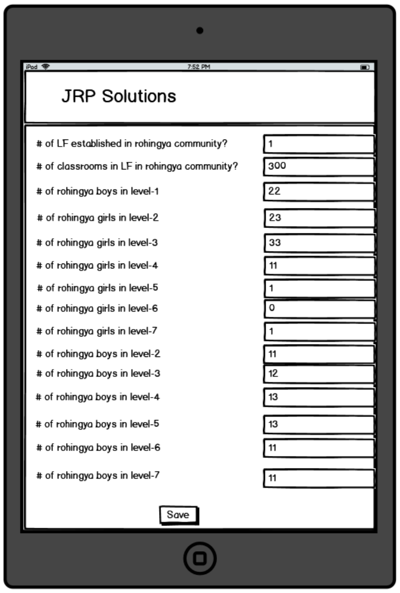


	Facility
☑	Brac learning Facility 1
☑	Brac learning Facility 2
□	Brac learning Facility 3

Once user is notified for data collection, there will be an option for the user to see which type of data grid data collection is required. After notified, user will download the new instance to the device.

User will see the instance for which data will be collected.





# of LF established in rohingya community?	1
# of classrooms in LF in rohingya community?	300
# of rohingya boys in level-1	22
# of rohingya girls in level-2	23
# of rohingya girls in level-3	33
# of rohingya girls in level-4	11
# of rohingya girls in level-5	1
# of rohingya girls in level-6	0
# of rohingya girls in level-7	1
# of rohingya boys in level-2	11
# of rohingya boys in level-3	12
# of rohingya boys in level-4	13
# of rohingya boys in level-5	13
# of rohingya boys in level-6	11
# of rohingya boys in level-7	11

If the user is assigned in multiple facilities, they can select the Facility for which data will be collected.

After Facility is selected, user will find the data collection form. Using which user will collect data.

User will be able to save the data before submitting the whole data to server side. User can make changes to the data before submitting to server-side publish

3.7. Frameworks

In the MIS system there needs to be a section for all the frameworks. There are multiple types of framework that should implemented in the system that will users to design and plan out their Program objectives. Frameworks are going to be used for settings up objectives, setting up baseline, settings targets, defining budgets, etc.

All the frameworks will be represented as grids. Users will be able to handle the data that they set in the grids can be linked with one another.

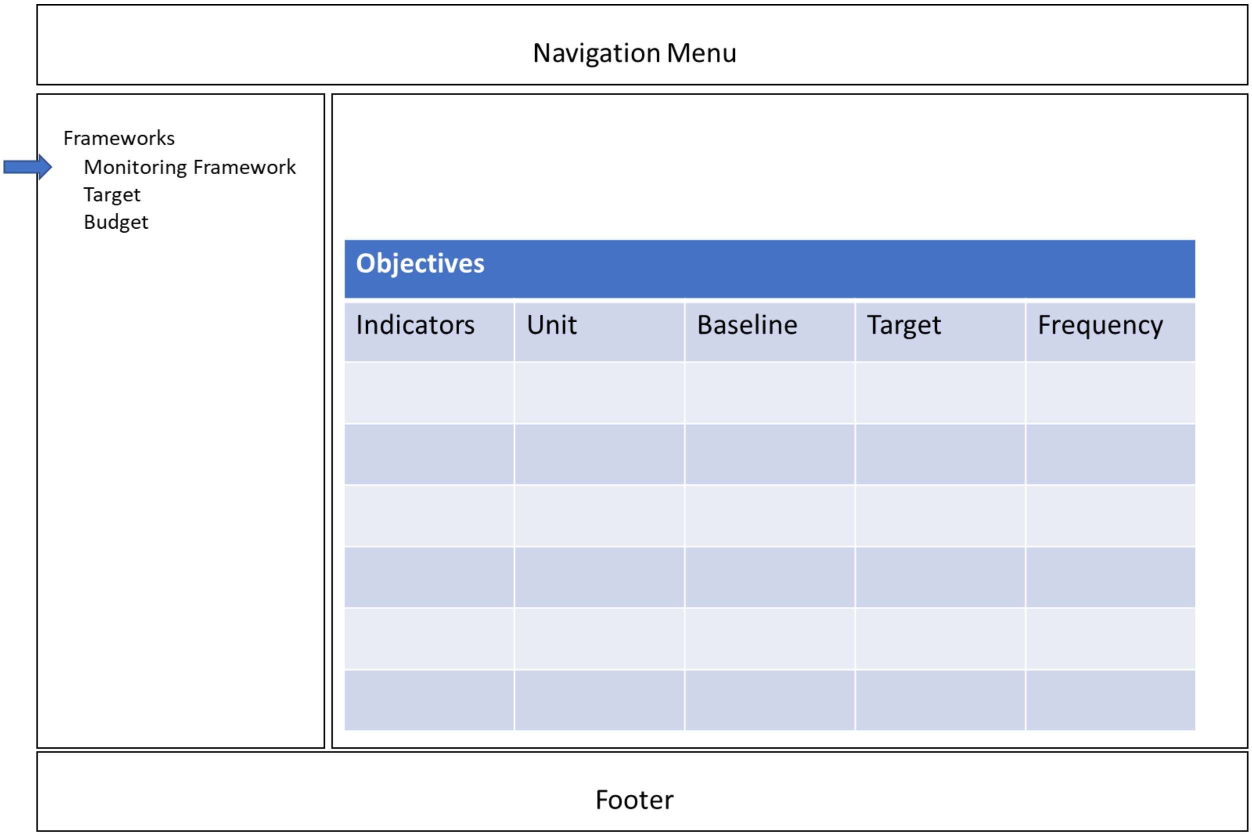
3.7.1. Different types of Frameworks

In the MIS, there needs to be a section for frameworks and all types of frameworks will be seen under the framework section. All framework will contain fixed columns. These fixed columns will be used in reporting and dashboards. Fixed Columns have relations with each other and will do mathematical functions. **These fixed columns will be provided by the UNICEF.** In system, the framework grid will be designed based on the provided column structures and relations.

User will be able to add more custom columns based to their needs. These custom columns will not be used in reporting and dashboard.

There three different types of framework required:

3.7.1.1. Monitoring Framework



Objectives

Indicators	Unit	Baseline	Target	Frequency

Monitoring Framework is a grid.

Using this grid, users will be able to plan.

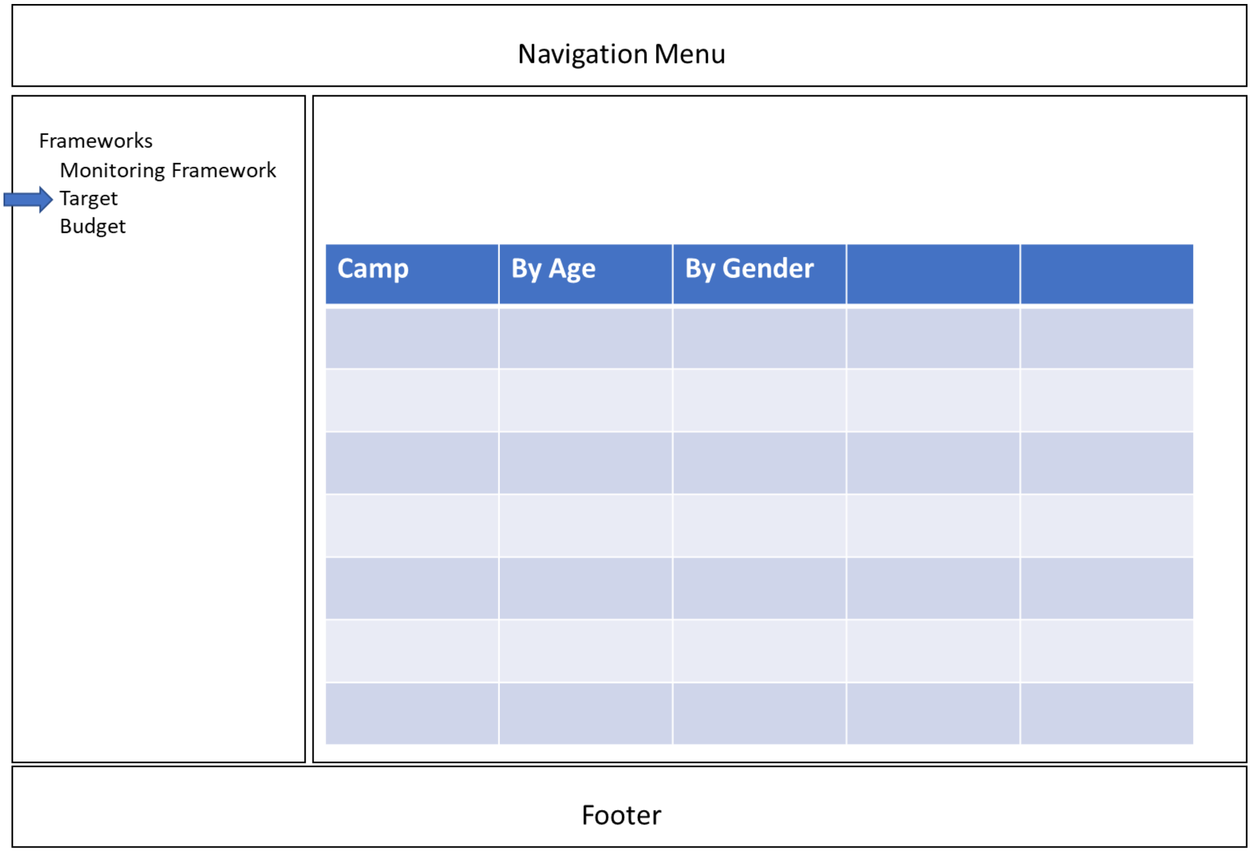
Based on this framework, users will be carrying out monitoring plan.

Users will be able to set objectives, indicators, unit, target, organization, frequency.

Under each objective, multiple indicators can be set.

For each indicator, other columns will be set, such as unit, baseline, target, data source.

3.7.1.2. Target Framework

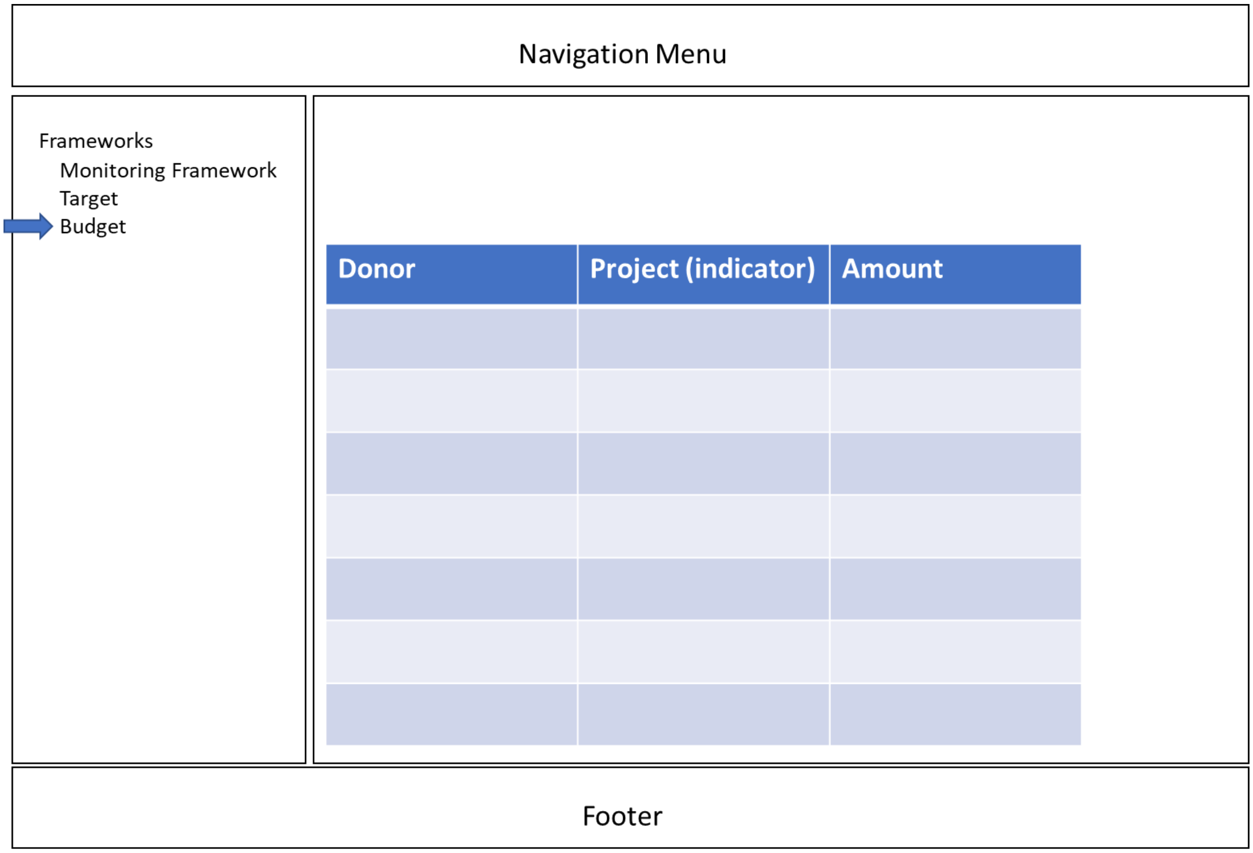


Camp	By Age	By Gender		

Using the Target Framework, users will be able to add target beneficiaries and set goals.

Targets are usually set against each camp.

3.7.1.3. *Budget Framework*

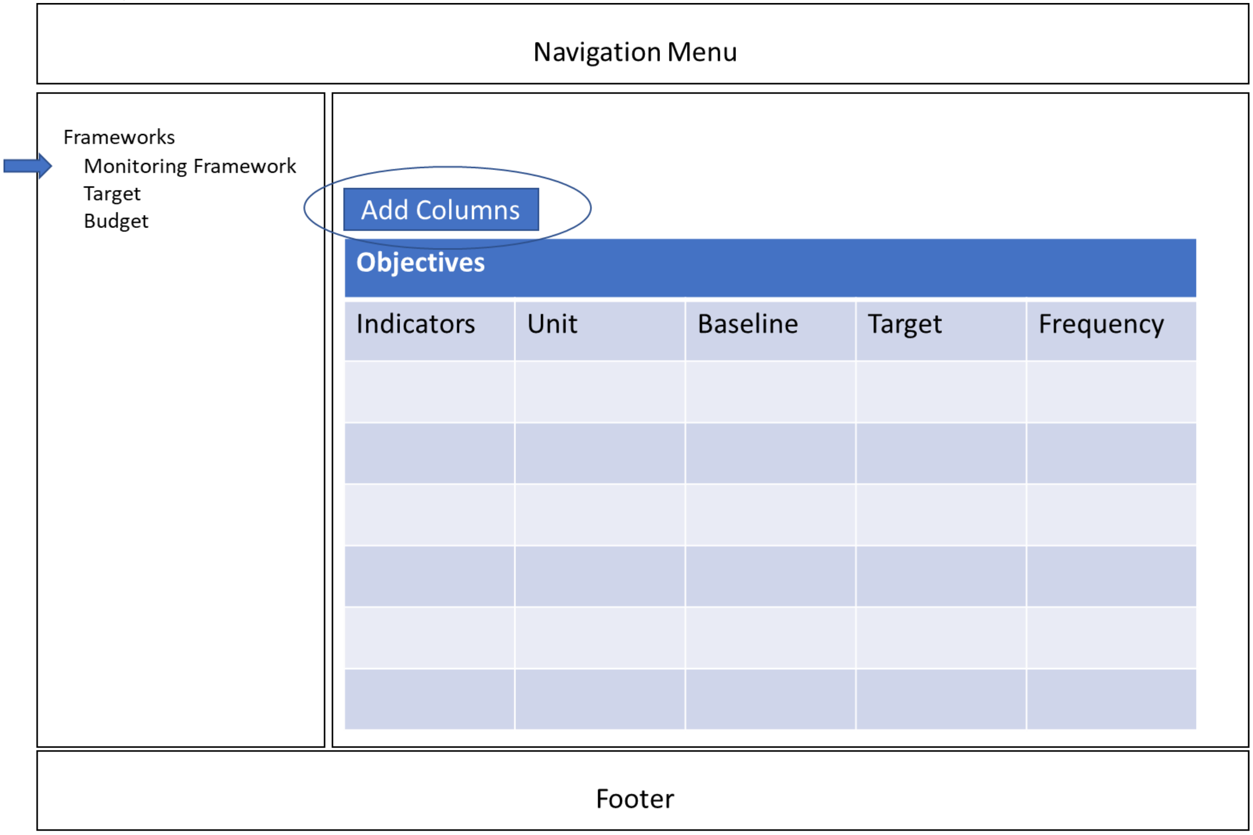


Donor	Project (indicator)	Amount

Users will be able to set budgets for each indicator, or activity, they run.

For that they will use the Budget framework.

3.7.2. Dynamic & fixed columns



Objectives				
Indicators	Unit	Baseline	Target	Frequency

All grids for frameworks will consist of some fixed columns (will be identified later) and some dynamic columns.

Users will be able to add new columns and set the type of data stored for those columns.

Types of columns are:

- Text
- Numeric
- Decimal
- Date
- Boolean
- List

Fixed columns will be used in reporting and dashboard.

Dynamic columns will be used for designing the frameworks.

3.8. Reporting

Conditions:

1. The reporting will be done upon the columns and calculations derived from the fixed columns. which exist in the system and are defined by UNICEF.
2. Necessary fields and the arithmetic operations with proper formatting and sample report files must be provided by UNICEF before development of the system begins.
3. Parameters used for generating reports and dashboard must be from columns in from data grids the system.

Based on the stored data on the system, reporting will be shown in the reporting part of the system. Users will be able to generate reports and customize the views of each report using the column turn on/off feature. Users will also be able to export the reports for other application can import the data exported from the MIS.

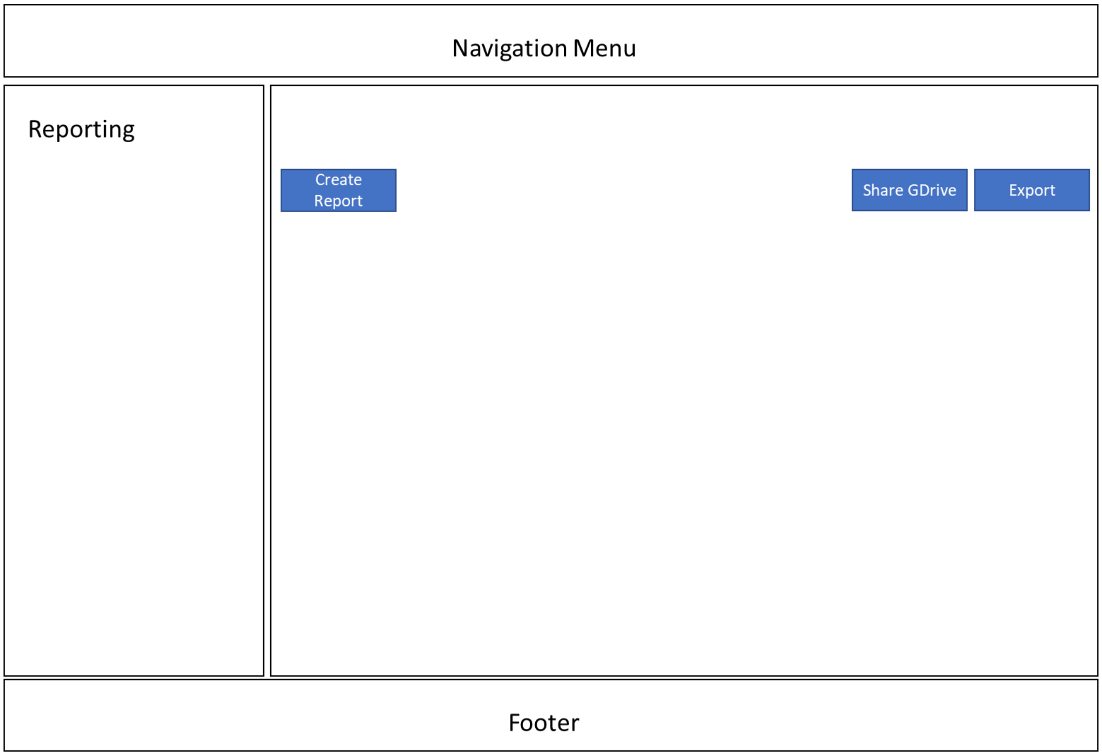
UNICEF will provide with the parameters that will be used in reporting based on the data populate in fixed columns from the grids in the system.

There will six different types of reports:

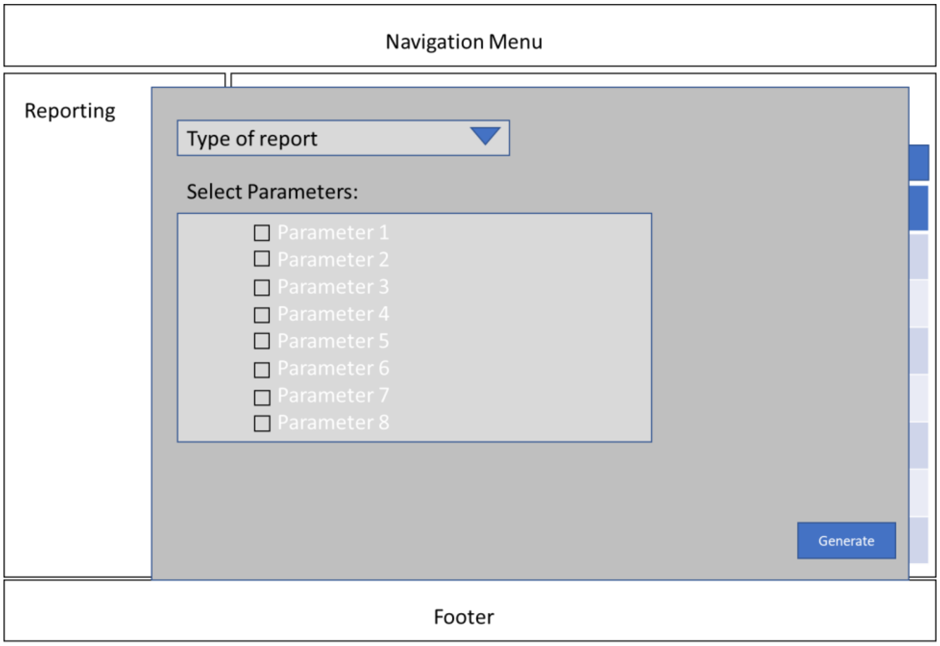
- 5W report
- Summary report
- Gap analysis report
- Damage report
- Camp wise report
- Duplication report

Reports will be generated from the Facility Grid Columns and the Beneficiary information that under each facility. For example: Beneficiary columns like Age, gender, Level will be calculated and will fill some of the defined columns for the reports.

Reports will be generated based on Fixed Columns from the data grids.



The user can select the defined 6 types of reports. Select the parameter provided and generate report.



Reports generated will be downloaded in Excel, CSV and shared on Google Drive.

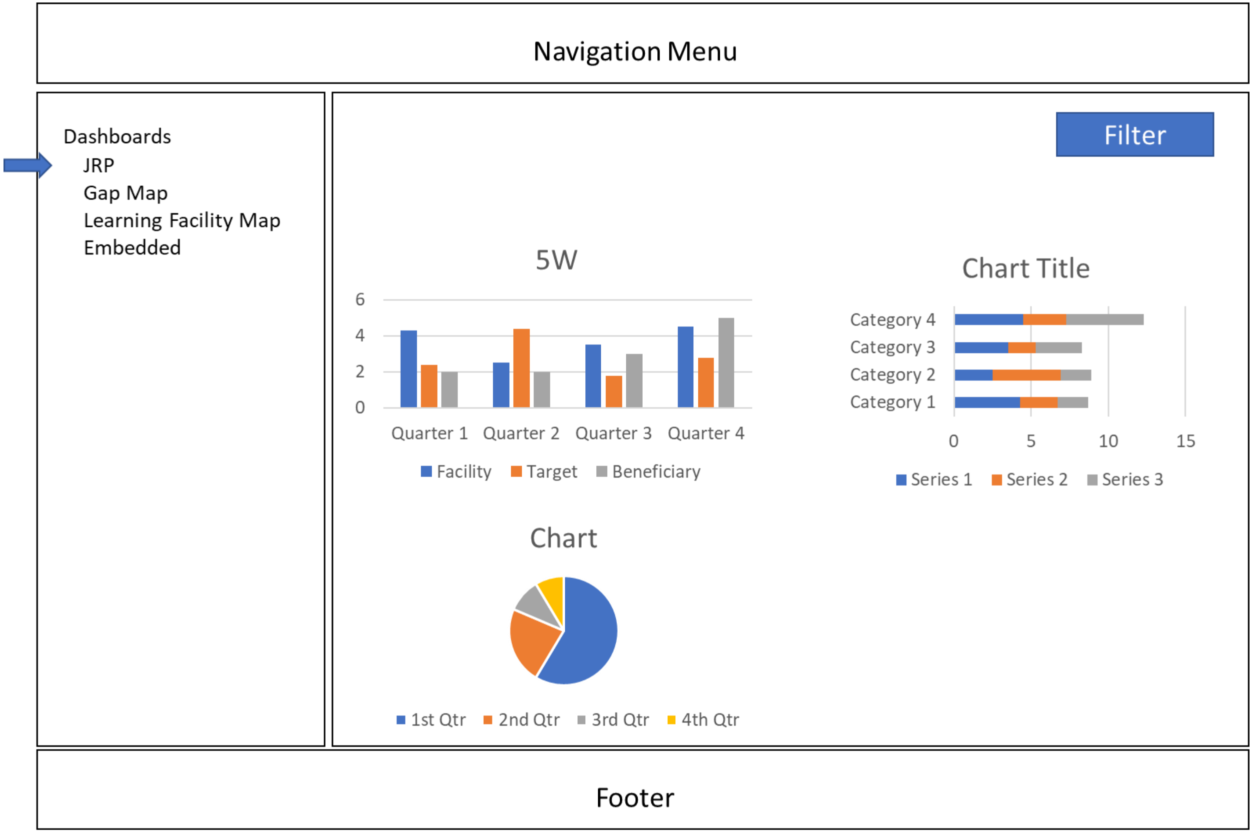
3.9. Dashboards

Conditions:

4. The reporting will be done only upon columns and calculations derived from the fixed columns. which exist in the system and are defined by UNICEF.
5. Necessary fields and the arithmetic operations with proper formatting and sample report files must be provided by UNICEF before development of the system begins.
6. Parameters used for generating reports and dashboard must be from the Fixed columns in from data grids the system.

This section describes the different types of dashboards that will be available to the system. Users will be able to generate dashboard and customize the views of each dashboard using the column turn on/off feature. Dashboards will be used by the users to get graphical reports using charts, graphs, and maps.

3.9.1. Aggregated dashboard



One of the main dashboards will be used is JRP.

Graphs populated in this dashboard will be based on the data stored in the Beneficiary Grid, Facility Grid, Monitoring Framework, Target Framework and Budget Framework.

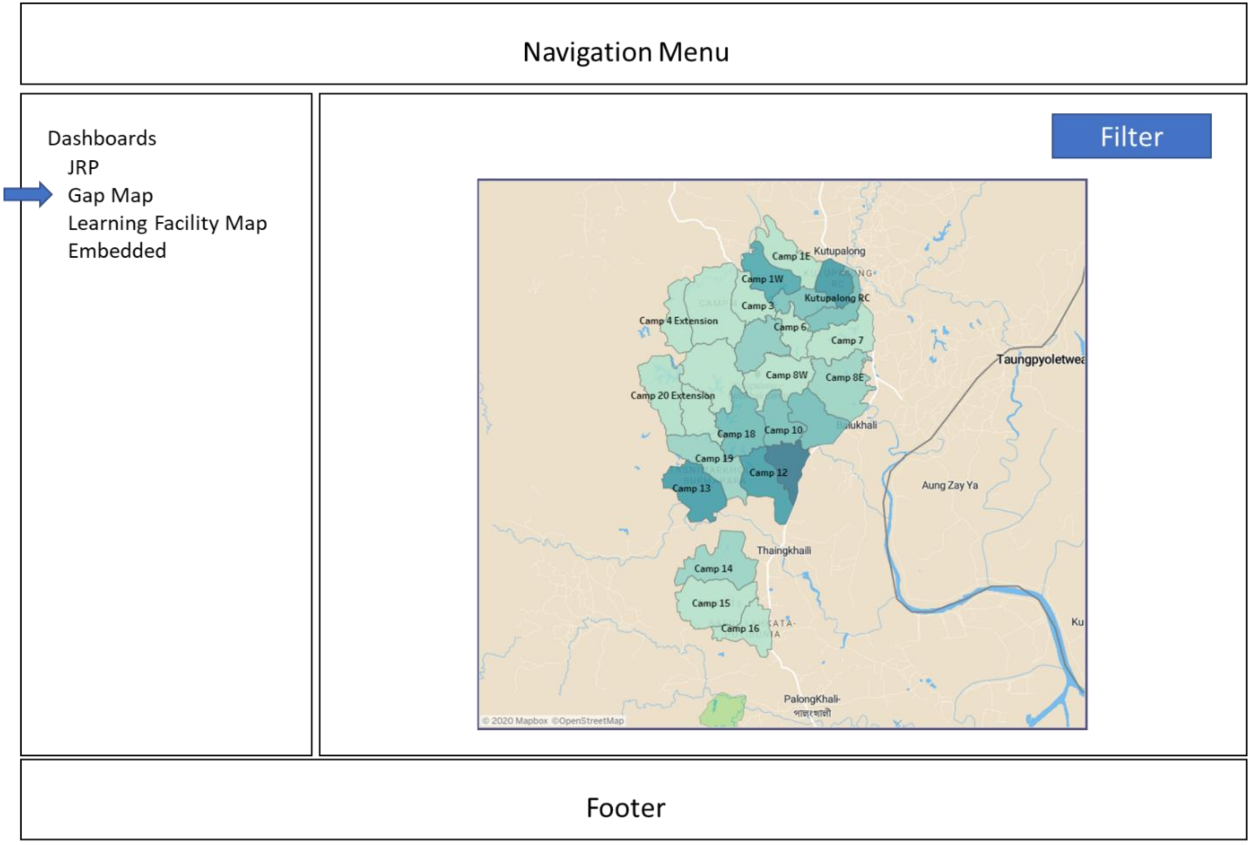
Users will be able to apply filters.

Filter will be done using the following sets of parameters:

- Implementing Partner
- Program Partner
- Camp
- Gender
- Disability
- Host/Refugee
- Age
- Level
- Time (instance)

[The above list was provided by UNICEF. They may request make edits to it later.]

3.9.2. Gap Map Dashboard



Navigation Menu

Dashboards

JRP

Gap Map

Learning Facility Map

Embedded

Filter

Footer

Gap analysis will be the dashboard that show different data in a map view.

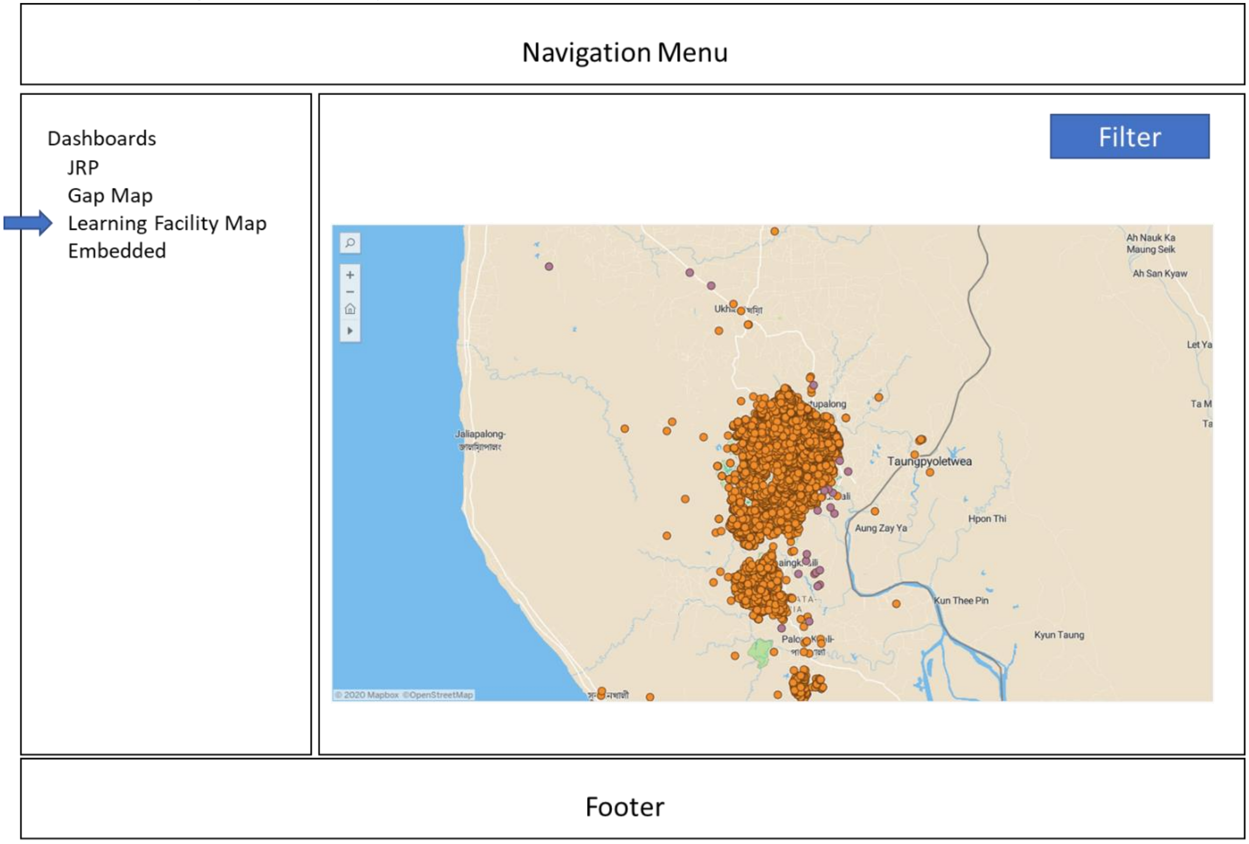
Users will be able to filter the map.

Filtering Criteria:

- Camp
- Gender
- Age range

Users will be able to export the map into PDF formats.

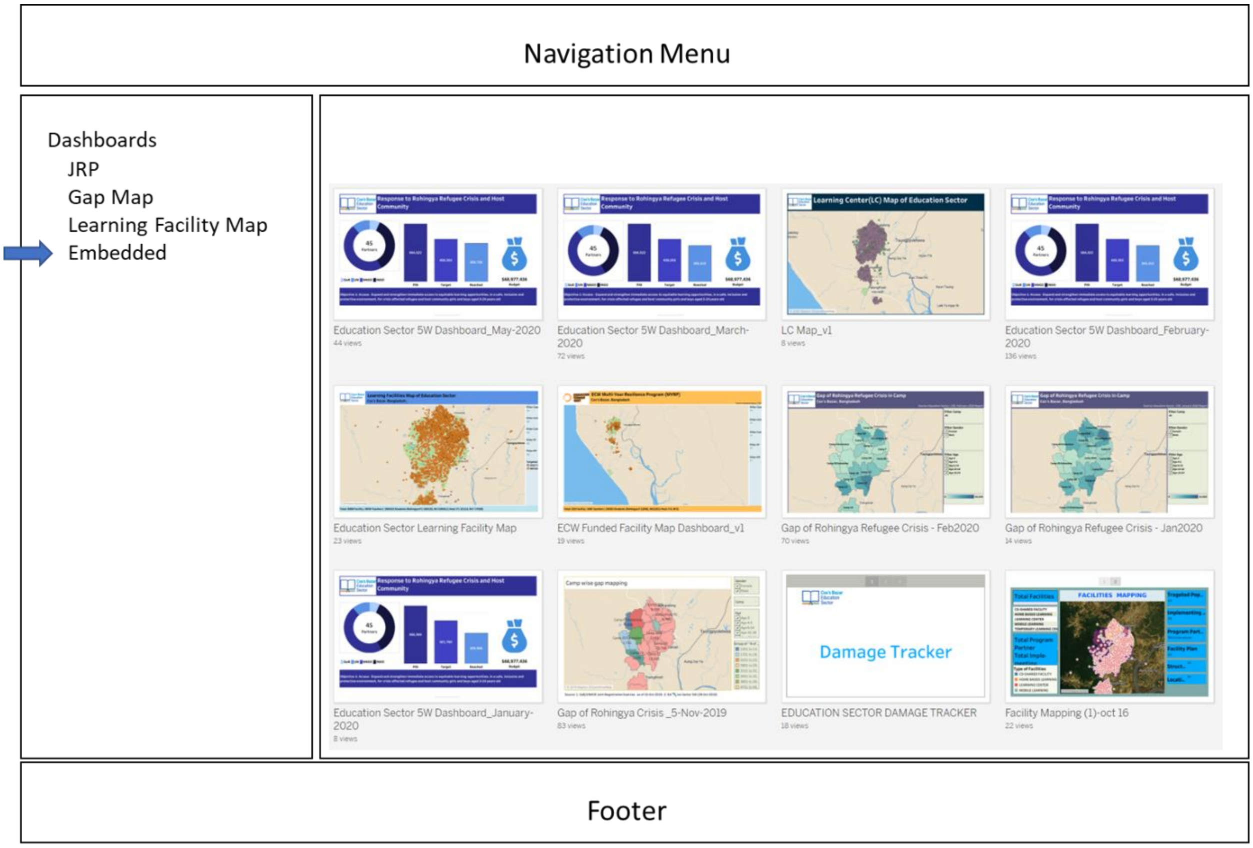
3.9.3. Learning Facilities Map Dashboard



Learning Facility map will be generated based on Location (Upazila & Union), Camp, Implementing Partners, Program Partners, and the number of Beneficiaries.

The purpose of this map is to provide a graphical representation of the locations of all the learning facilities under specific camps, and the number of beneficiaries assigned to them.

3.9.4. Embedded Dashboard


Navigation Menu
Dashboards
JRP
Gap Map
Learning Facility Map
Embedded
Education Sector 5W Dashboard_May-2020
44 views
Education Sector 5W Dashboard_March-2020
72 views
LC Map_v1
8 views
Education Sector 5W Dashboard_February-2020
136 views
Education Sector Learning Facility Map
23 views
ECW Funded Facility Map Dashboard_v1
19 views
Gap of Rohingya Refugee Crisis - Feb2020
70 views
Gap of Rohingya Refugee Crisis - Jan2020
14 views
Education Sector 5W Dashboard_January-2020
8 views
Gap of Rohingya Crisis _5-Nov-2019
83 views
Damage Tracker
EDUCATION SECTOR DAMAGE TRACKER
18 views
Facility Mapping (1)-oct 16
22 views
Footer

The dashboard should also consist of an embedded interface from a 3rd party provider.

System will expose the API so it can be used in 3rd party application.

The 3rd party application Web URL will be embedded by (example: iframe) into the system.

Note: System will expose reasonable number of APIs to allow embedded dashboard. And APIs will be in one specific standard. The application will not take responsibility for the 3rd party application's performance or any kind of malicious activity.

Types of vulnerability that may occur:

1. UI of the system may break when displaying the 3rd party dashboard.
2. The system may become vulnerable to cross-site attacks.
3. If the 3rd party is attacked with malicious entity that might put the system at risk.
4. Possibility of phishing attacks on the system via the 3rd party server.

4. Non-Functional Requirements

4.1. System Architecture and Security

The application will be structure in the client server architecture. Client refers to the Web Application (Front End). Server refers to the API server (Back End). In general, front end works through the backend API. To provide the offline functionality, a caching mechanism will be provided on the client end, so that user don't need to be dependent on the backend and whenever Internet communication is available, the frontend will automatically trigger the sync for both ends (See Section 4.2). The server application will contain all the business logic and the actual data in a database. The client side will be web browsers in desktop/laptop computers. The web portal will be built responsive to enable viewing comfort various display sizes, including mobile browsers. The application will be developed operating system independent so that it can be deployed and maintained either on Linux server or Mac OS or Windows. The system will be deployed on Linux server to avoid any additional licensing.

The server-side of the application will be developed using C# language using .Net core platform and the front end will developed using a JavaScript platform based (e.g. Angular.js) for provide the optimization. For graphs and chart generation, open-source libraries will be used.

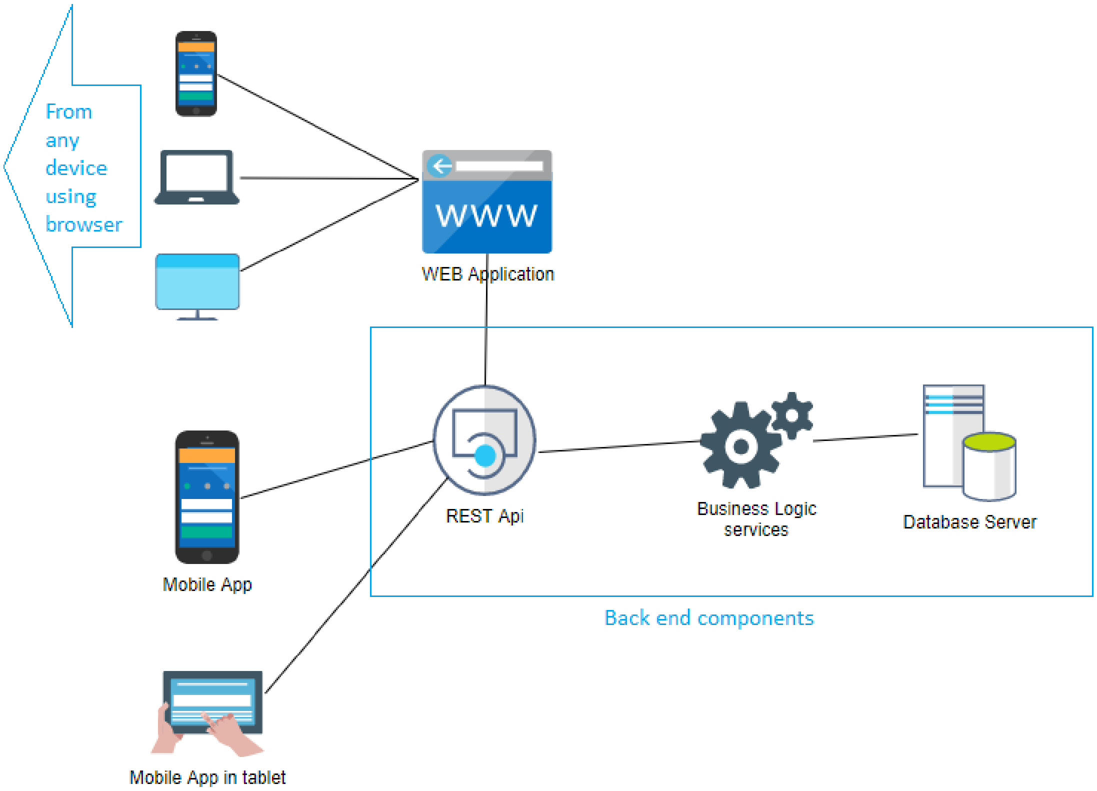
The database for the application will be developed and designed with MySQL. The server side will be exposed with REST API interfaces. The data transfers will also be minimal using JSON to keep the overall communication low. REST interface API will provide services to the Android based application.

The Android application will be built with native code.

There should be security levels for the access to the data as not all levels of users will have access to the same data. For authentication and security, JWT token-based authentication mechanism will be used for it is robust, secured and easily extensible. Will have to provide client-end data storage upto a certain period and level for offline data collection feature, which will be synced bi-directional once internet connection is available. For all users' data, a standard encryption will be implement based on the organization's privacy-policies. The standard encryption should also be able to project sensitive data in case of an accidental exposure of the database and the users' sensitive data cannot be read without the decryption logic.

For both Web based and Mobile base application usage, there will be a user login that will provide security to the application and user data. Users with various permissions and roles will be able use the application according to their rights.

Data from the database can be accessed using RESTful APIs which will be authorized via access tokens. Raw database will not be exposed. Users with need will be able to access fixed and dynamic columns data through these APIs.



4.2. Offline usage

Kaz Software will provide the offline data storage mechanism for the Android application, so that the app user can fill the forms data upto 1-2 days without need of Internet connection. However, once the user connects the device to an internet connection, then they will have the option to sync the data with the backend of the application.

To support cross platform data collection, Kaz Software will provide a JavaScript/HTML based web application like the android app which needs to run over internet to initialize in the browser of your device. Later, if you want to collect data without the internet, the same web application can work from your browser cache. Finally, to sync to/from backend server you need to load the web app in your browser again when you have internet connection. We should remember that we will not delete browser cache in the meantime, otherwise offline data will be destroyed.

4.3. Data Validation

The system will include a validation check on the data imported and stored. The excel spreadsheet has no validation mechanism. A data validation will be implemented in system to check data integrity and elimination of major errors. On the server-side, there needs to be an activity logging, a mechanism which will help to revise the previous version of the data to backtrack and audit control. If the system finds erroneous data, a mechanism should be implemented which will revert it.

4.4. Quality Control

Quality control will be done at several levels. Each storyboard will go through an agile testing process, which has been widely adopted in the software industry as the most complete and responsive testing scheme. The agile testing process follows the following steps:

4.4.1. Developer desk testing

Once a story or part of a story is completed an informal testing session will done at the developer's own desk where both the developer and the QC (quality control) engineer will walk through the story to identify issues at the surface level. An emulator will be used this stage for running the tests. This agile testing process shortens the QC process by identifying issues early on the development process.

4.4.2. Integration testing

Once the issues found at the developer's desk are resolved the QC will then create a build to run the integration tests. The goal of this phase is to check if the features developed works with the rest of the features in the application.

4.4.3. Automated or semi-manual regression testing

A testing regime will be setup for running automated or semi-manual regression testing of builds. The aim for automating the process will be to get to issue faster on newly created issues where the features were working in previous builds.

4.4.4. Usability testing

Testing will be setup with user groups (selected from target user groups, e.g. researchers) to test the usability of the application when the application has reached a point where some of its features are usable.

5. Conclusion

References

All the requirement and features explained and covered in this document was based upon the Terms of Reference document presented by UNICEF, discussion on conferencing platform Zoom.

Disclaimer

This document will be treated as a contract of features that will be developed by Kaz Software and UNICEF. Any changes that will be requested in this document will, or addition of features, must come through a change request and agreed by both the parties. Changes will be documented in this document and shared between both the parties for approval.

Timeline, estimation, and delivery of the project will be documented in a separated agreement. Feature changes addressed in this document may also impact on the timeline, estimation, and delivery for this project for that both the parties must come an agreement, may be in separated document.