VISIDA App Design Documentation

Introduction

Purpose

The purpose of this document is to describe, in detail, the overall design of the VISIDA Android application developed as part of the VISIDA Project.

Scope

The VISIDA Android application is intended to be used by researchers of the university to collect voice and image data to be processed further to provide an accurate dietary profile. The Android app along with various other tools developed as part of the project will provide means for quickly and easily providing highly accurate dietary profiles, particularly for low and middle income communities.

Overview

This document has the following structure:

- System overview to explain what the Android application is and how it fits amongst the VISIDA system
- System Architecture to provide detailed description of the VISIDA Application architecture and a high level description of how it fits within the larger VISIDA System architecture
- Data Design which describes all the data collected and used within the VISIDA Application
- Component Design provides a detailed description of the VISIDA Android Application's structure
- Human Interface Design provides a description of all the human interfaces within the VISIDA Android application
- Requirements
- Appendices

Reference Material

Id	Document	Location	Description	Notes
1	VISIDA_Processes	Repo/Documents	Describes the process the user goes through to achieve each use case	Branch: master
2	VISIDA App_Tech	Repo/Documents	Gives a more detailed description of the app and it's internal workings.	Branch: release/configurable

3	Language_Extension_Guide	Repo/Documents	Describes the process to add another language to the app.	Branch: release/configurable
4	VISIDA_Strings_Master	Repo/Documents	Contains all of the strings and their translations in Khmer and Swahili. This includes Audio Buttons and Instructions	Branch: master
5	OPP1171389__2017_Newcastle_Proposal_Narrative_Final	Owncloud/VISIDA project/application	Project application.	

Definitions and Acronyms

Acronym	Definition
VISIDA	Voice Image Sensor technologies for Individual Dietary Assessment
CMS	Content Management System
App	Application
LMIC	Low and lower-middle income country
LTE	Long-Term Evolution
MVVM	Model View ViewModel
JSON	JavaScript Object Notation

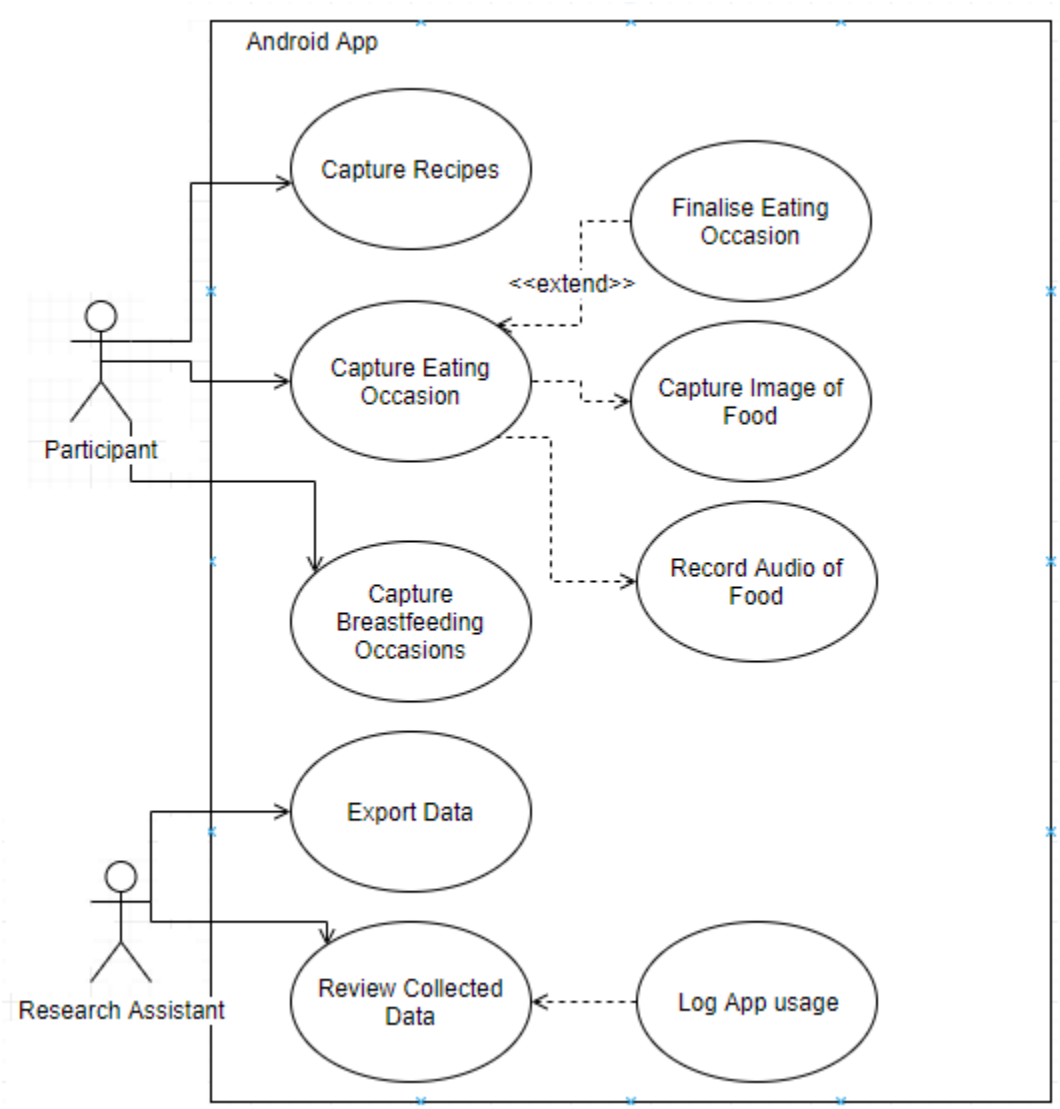
System Overview

From the proposal narrative found in document ***5***: "*Within low and lower-middle income country (LMIC) settings, regular and comprehensive dietary assessment at the individual level is challenging due to resource intensive methods which rely on skilled research staff and observation".*

The VISIDA android application aims to "*Standardize the collection of dietary intake information and improve data quality at the individual level including meal preparation information and shared meals/plates using active (smartphone image-voice food record) capture"*.

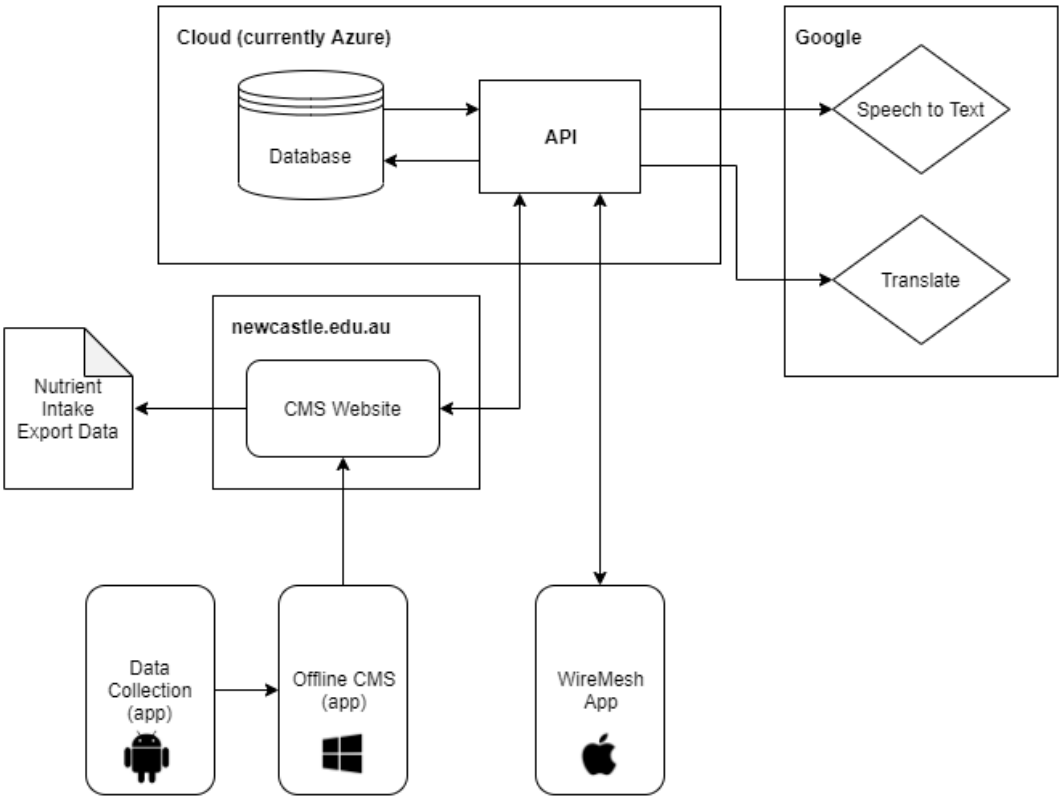
The app will collect data by capturing and storing images and audio recordings about an individuals dietary intake over a period of time. This data is then to be processed further by other tools developed for this project.

The android Application aims to achieve the following use cases:



System Architecture

Architectural Design



The core of the system is the cloud based CMS which is composed of an SQL database and Web API. This CMS uses Google API's to support translation and speech to text functionality. Consuming the API is the frontend web application which provides an interface to the user. These products combine to make up the primary data processing components of the system.

Supplementary to the core CMS are the various mobile applications, including the subject of this document, the VISIDA Android Application for data collection. The CMS consumes the data collected by the VISIDA Android app and presents it to the user for further processing.

The application functions 'offline' and the data is exported locally to the device and uploaded manually into the CMS. This allows users to use the device in areas with no LTE signal and incurs no additional data costs.

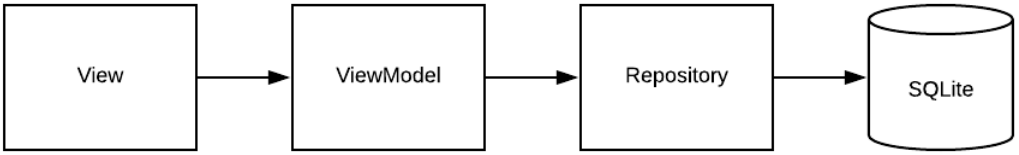
There also exists the option of importing the data into an "Offline CMS" which provides a much more basic, trimmed down version of the CMS to review data offline. This Offline CMS is a windows application developed to be mobile compatible and used 'in the field'.

As part of the toolkit to help users of the VISIDA application provide high quality dietary profiles an iOS app has been developed which can be used to help estimate the volume of (food) items in an image. This application requests images from the cloud based CMS image and returns the volume estimate provided by the user.

Decomposition Description

Only the VISIDA Android Application will be discussed here as it is the primary focus of this document.

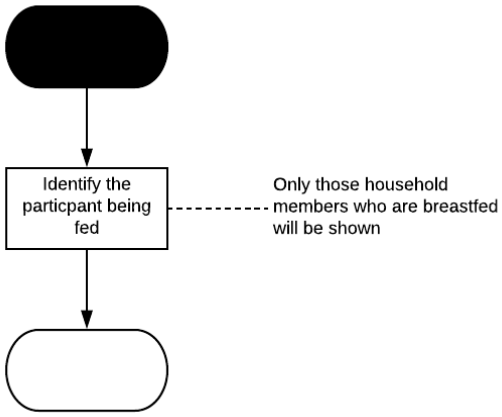
The overarching application architecture is based on Android best practices and the Model View ViewModel (MVVM) pattern:



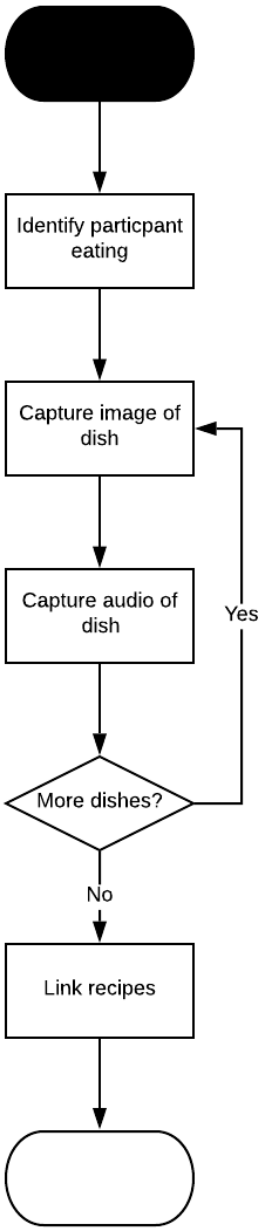
This architecture pattern is supported by Android architecture components. It provides a clean architecture which de couples the data and business logic from the interface allowing for rapid interface development cycles with little impact on the underlying business logic.

The following are high level flow charts outlining the basic process to achieve the four key use cases:

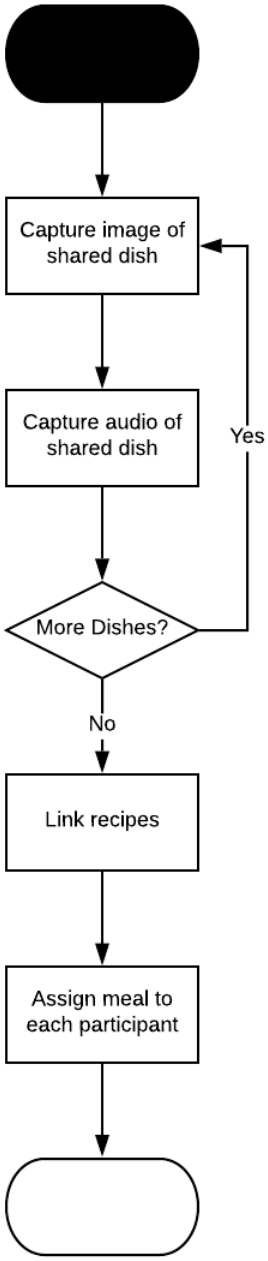
Breastfeed



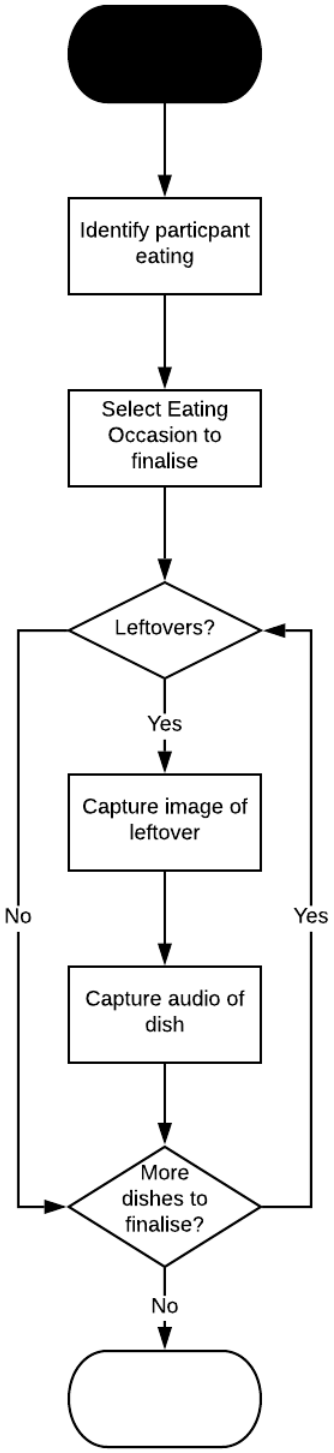
Eat - Individual



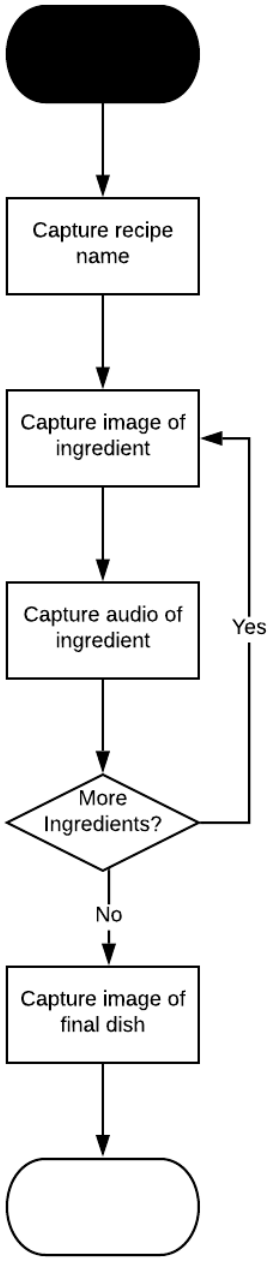
Eat – Shared



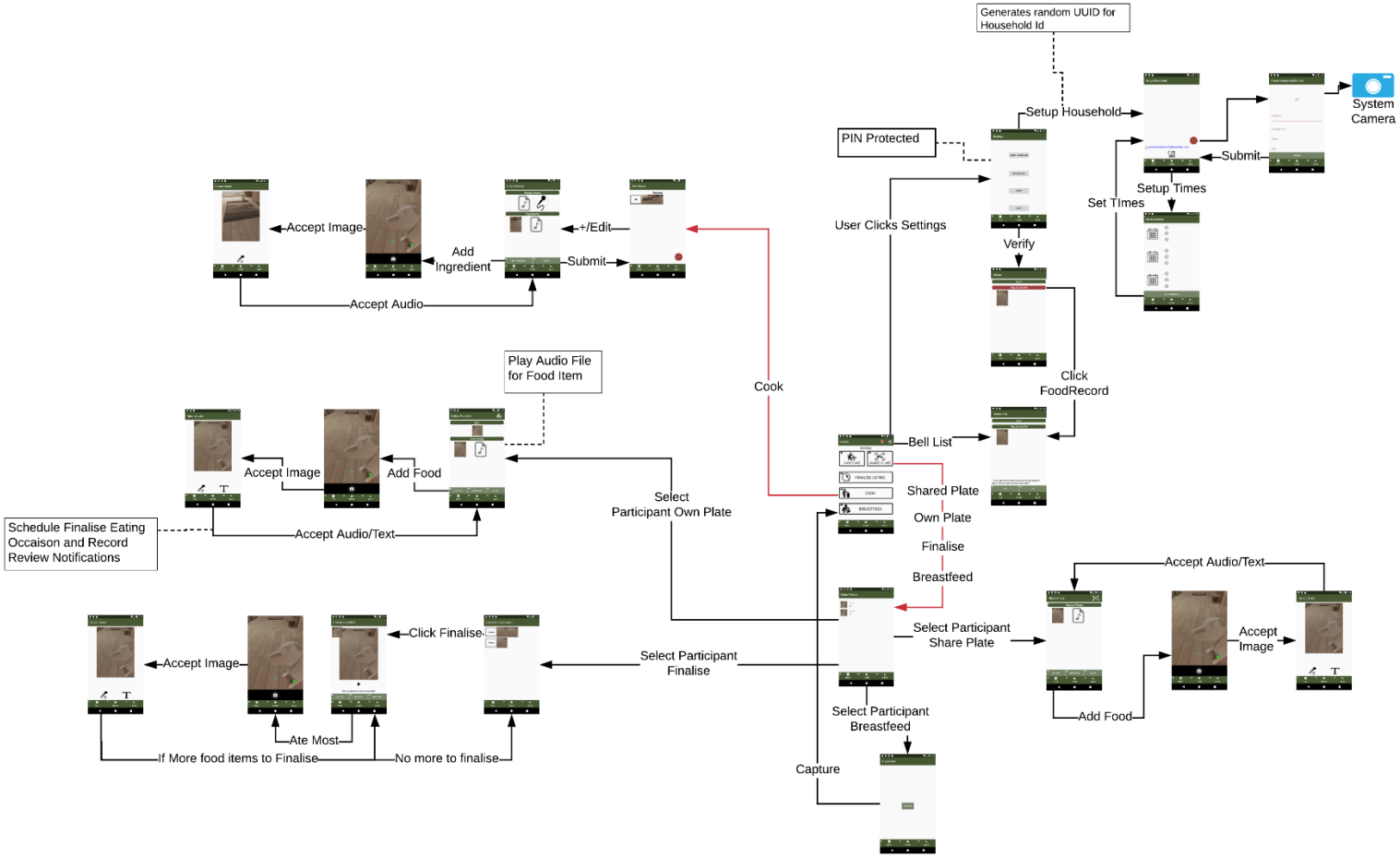
Finalise Eating Occasion



Cook



The whole applications process flow is described below with screenshots of the current iteration. This diagram provides another representation of the previous flowcharts and explains how some common functionality such as Selecting a Household Member is shared amongst the processes.

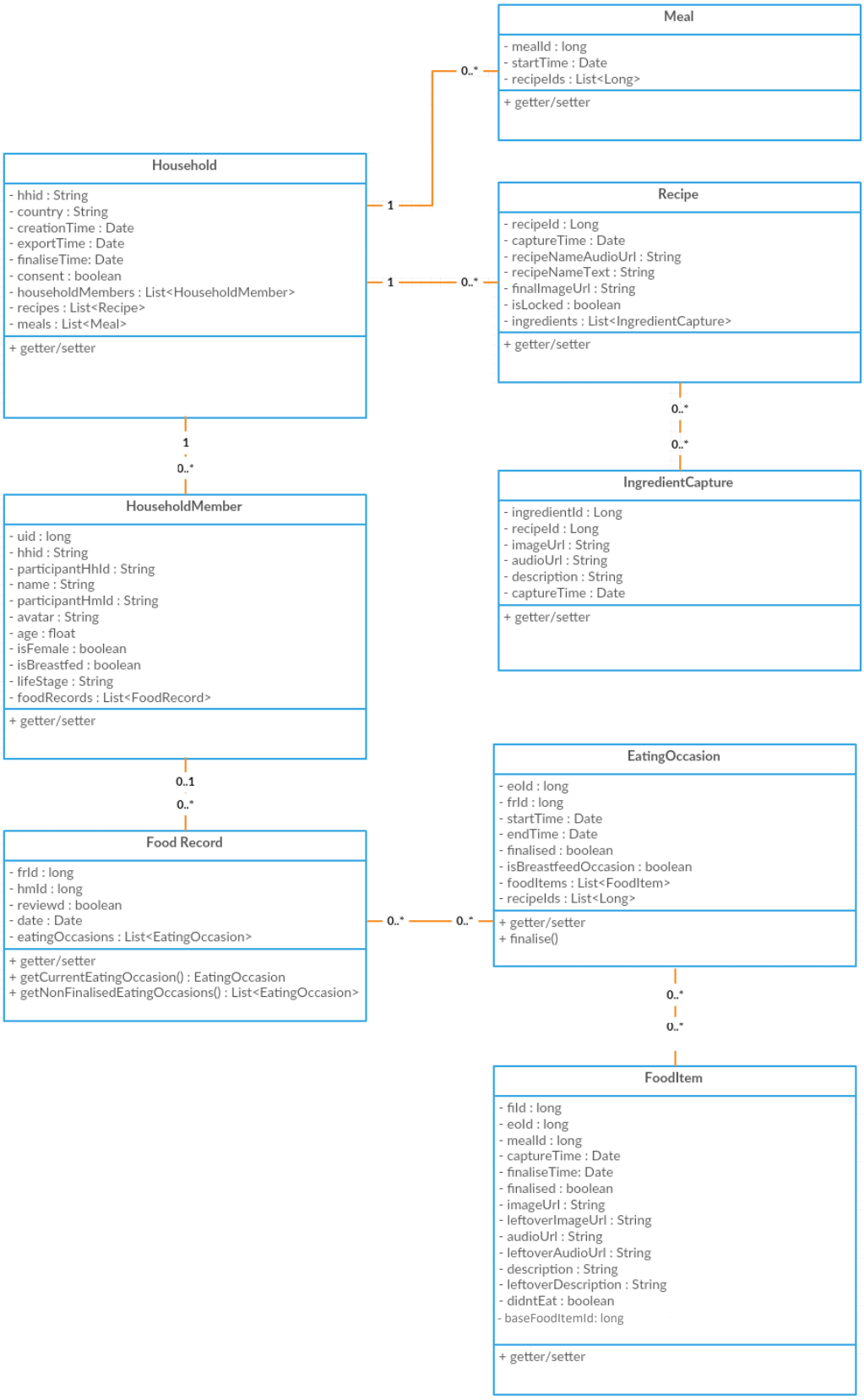

Generates random UUID for Household Id
Setup Household
System Camera
Submit
Setup Times
Set TImes
PIN Protected
User Clicks Settings
Verify
Click FoodRecord
Bell List
Shared Plate
Own Plate
Finalise
Breastfeed
Cook
Accept Image
Add Ingredient
+/Edit
Submit
Accept Audio
Play Audio File for Food Item
Accept Image
Add Food
Accept Audio/Text
Schedule Finalise Eating Occaison and Record Review Notifications
Select Participant Own Plate
Select Participant Finalise
Select Participant Share Plate
Select Participant Breastfeed
Capture
Accept Audio/Text
Accept Image
Add Food
Accept Image
Click Finalise
Ate Most
If More food items to Finalise
No more to finalise

Data Design

Data Description

The following diagram describes the data which is collected and stored within the VISIDA android application and its internal relationships. These entities are stored locally on the device in an SQLite3 database. Upon export the data structure is serialized into JSON and zipped up along with the image and audio files to be uploaded into the CMS.

Images are stored as “.jpg” and Audio Files are recorded as “.mp3”.


Meal
- mealId : long
- startTime : Date
- recipeIds : List<Long>
+ getter/setter
Household
- hhid : String
- country : String
- creationTime : Date
- exportTime : Date
- finaliseTime: Date
- consent : boolean
- householdMembers : List<HouseholdMember>
- recipes : List<Recipe>
- meals : List<Meal>
+ getter/setter
Recipe
- recipeId : Long
- captureTime : Date
- recipeNameAudioUrl : String
- recipeNameText : String
- finalImageUrl : String
- isLocked : boolean
- ingredients : List<IngredientCapture>
+ getter/setter
IngredientCapture
- ingredientId : Long
- recipeId : Long
- imageUrl : String
- audioUrl : String
- description : String
- captureTime : Date
+ getter/setter
HouseholdMember
- uid : long
- hhid : String
- participantHhId : String
- name : String
- participantHmId : String
- avatar : String
- age : float
- isFemale : boolean
- isBreastfed : boolean
- lifeStage : String
- foodRecords : List<FoodRecord>
+ getter/setter
Food Record
- frId : long
- hmId : long
- reviewd : boolean
- date : Date
- eatingOccasions : List<EatingOccasion>
+ getter/setter
+ getCurrentEatingOccasion() : EatingOccasion
+ getNonFinalisedEatingOccasions() : List<EatingOccasion>
EatingOccasion
- eoId : long
- frId : long
- startTime : Date
- endTime : Date
- finalised : boolean
- isBreastfeedOccasion : boolean
- foodItems : List<FoodItem>
- recipeIds : List<Long>
+ getter/setter
+ finalise()
FoodItem
- fiId : long
- eoId : long
- mealId : long
- captureTime : Date
- finaliseTime: Date
- finalised : boolean
- imageUrl : String
- leftoverImageUrl : String
- audioUrl : String
- leftoverAudioUrl : String
- description : String
- leftoverDescription : String
- didntEat : boolean
- baseFoodItemId: long
+ getter/setter
1
0..*
1
0..*
1
0..*
0..*
0..1
0..*
0..*
0..*
0..*
0..*

Data Dictionary

Entity	Data	Type	Description
Eating Occasion	eoId	Long	Unique Id of an Eating Occasion
	frId	Long	Foreign Key id of the parent Food Record
	startTime	Date	Date of when the Eating Occasion was created
	endTime	Date	Date of when the Eating Occasion is finalised
	Finalised	Boolean	Flag to show if the Eating Occasion has been finalised or not
	isBreasfeedOccasion	Boolean	Flag to indicate if the eating occasion is capturing a breastfeeding occasion
	foodItems	List<FoodItem>	List of all the children Food Items attributed with this Eating Occasion
	recipeIds	List<Long>	List of all the foreign Key recipe Ids attributed to this Eating Occasion
Food Item	fiId	Long	Unique Id of Food Item
	eoId	Long	Foreign Key Id of parent Eating Occasion
	mealId	Long	Foreign Key Id of associated meal data. This is set if the food item was captured as part of a Shared Dish.
	captureTime	Date	Time of Food Item's capture
	finaliseTime	Date	Time of when food item was finalised
	Finalised	Boolean	Flag representing if the food item is finalised

	imageUrl	String	Device Local URL of the image taken for this food item
	leftoverImageUrl	String	Device local URL of the image take for the leftovers (if any) for this food item.
	audioUrl	String	Device Local URL of the audio recorded for this food item
	leftoverAudioUrl	String	Device local URL of the audio recorded for the leftovers (if any) for this food item
	didntEat	Boolean	Flag representing if this food item was eaten. This is set to true if the food item is shared and the participant did not eat any.
	baseFoodItemId	long	Foreign Key of the original shared dish food item used to create an individual copy for each household member.
Food Record Household	frId	Long	Unique Id of the Food Record
	hmId	Long	Foreign Key Id of the Parent Household Member
	Reviewed	Boolean	Flag indicating if the food record has been reviewed.
	Date	Date	Date when the food record is created
	eatingOccasions	List<EatingOccasion>	List if children Eating Occasions associated with this food record.
	Hhid	String	Unique Id of the household
	country	String	Name of country the study is taking place in
	creationTime	Date	Date/Time of household creation

	exportTime	Date	Date data was exported
	finaliseTime	Date	Date/Time the user would like reminders to finalise their days to be sent.
	Consent	Boolean	If the user consents to their data being used by third party tools (Google Speech to text)
	householdMembers	List<HouseholdMember>	List of participating household members
	Recipes	List<Recipe>	List of recipes created by this household
	meals	List<Meal>	List of meals eaten at this household
HouseholdMember	uid	Long	Unique Id of the household member
	Hhid	Long	Foreign Key Id of the parent Household
	participantHhId	String	Study wide unique identifier for the household. Provided by research assistants at setup
	Name	String	Name of the household member
	particpantHmId	String	Study wide unique identifier of the household. Provided by research assistant at setup
	Avatar	String	Device local URL of the image used for the avatar of the household member
	Age	Float	Age of the household member
	isFemale	Boolean	Flag if the household member is a mother (will be breastfeeding)
	isBreastfed	Boolean	Flag if the household member is currently being breastfed. Only available if the age of

			the household member is <= 4
	Lifestage	String	Description of the lifestage of the household member: Current possible values are: None, Pregnant, Breastfeeding, Pregnant & Breastfeeding.
	foodrecords	List<FoodRecord>	List of all the food records captured for this household member
IngredientCapture	ingredientId	Long	Unique Id of the Ingredient
	recipeId	Long	Foreign Key id of the parent Recipe
	imageUrl	String	Device local URL of the image of the ingredient
	audioUrl	String	Device local URL of the audio file description
	Description	String	Text description of the ingredient
	captureTime	Date	Date/Time of when the ingredient was captured.
Meal	mealID	Long	Unique id of the Meal
	startTime	Date	Date/Time the meal was created
	recipeIds	List<Long>	List of Foreign Key Id's of the recipes eaten during the meal
Recipe	recipeId	Long	Unique id of the Recipe
	captureTime	Date	Date/Time of when the Recipe was created
	recipeNameAudioUrl	String	Device local URL of the audio file containing the recipe name.
	recipeNameText	String	Recipe Name text

	finalImageUrl	String	Device local URL of the image of the finished recipe.
	isLocked	Boolean	Flag if the recipe has been locked
	ingredients	List<IngredientCapture>	List of child ingredients that are part of this recipe.

Component Design

Since the main high level use cases for the app (EAT, FINALISE, COOK, BREASTFEED) are described graphically and their functionality they won't be further broken down here.

Image Capture

The image capture is handled by the "TakePhotoFragment" within the "CameraActivity". These work together by having the Activity pass the name of the image to the fragment, in some cases this name is just a format to be filled in later after the image has been captured. The camera activity will return the name of the image taken in the result intent (AppConstants.IMAGE_NAME). It is the calling activities responsibility to handle the image name, that is, the calling activity must know that whether the resulting image name will be completed or still require formatting.

The image capture provides a preview of the captured image with a Tick or Cross for the user to use or discard the taken image. Once the preview has been accepted the image will be saved to the device and image name returned to the calling activity.

Audio Capture

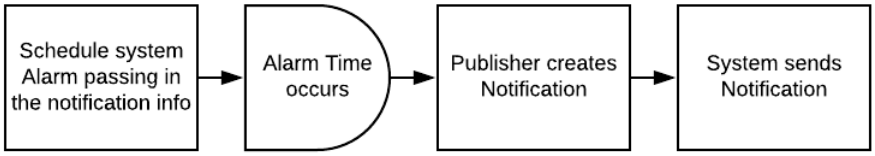
The audio capture is handled by the "AudioRecordingFragment" within the "AudioActivity". This component accepts the image name/url to display the image to the user while they describe it's contents. It a;so accepts the file name of the audio file. The audio is recorded and stored on the device. The user is then able to listen back to the recording to verify it. If they accept the recording the activity returns the file name of the audio recording (AUDIOFILE_NAME).

It is up to the calling activity to manage the file name of the audio recording. The activity/fragment does not manipulate the file name during the recording process.

Notification

The notification system uses Android's Notification API to handle the delivery of push notifications and the Alarm API to handle the scheduling of notifications. There are numerous types of notifications: Daily Reminders to wear the wrist sensor, daily reminders to collect data (scheduled for Breakfast, Lunch and Dinner), Reminders to finalise Eating Occasions if they have not done so already and a reminder to review the day. The Eating Occasion and Record Review notifications are stored in the database.

The process for setting a notification is as follows:



Instructions

The instructions module allows for both image and video instructions to be displayed in any language that a translation is provided for. The Instruction class contains the resource Id of the text and media to display in the instruction dialog.

When the instructions are opened for an activity, the resource id's are looked and the media resource's type is determined (image: 'drawable', video: 'raw') and the appropriate instruction fragment is loaded with the data.

Since the resources are loaded at run time, any localisation settings will automatically take care of translations provided there exists an "instructions_text.xml" and "instructions_media.xml" file for the desired language.

Meals

When the app is configured to handle shared dished an extra button is available to capture shared dishes. When a shared dish occasion occurs a Meal is created and all the image/audio recordings are stored in a special "EatingOccasion" with the meal Id set to this meal.

Once the finalise button is pressed a simple algorithm is run to create individual eating occasions for each participant containing all the image/audio recordings. The image and audio files are not duplicated.

The algorithm is as follows:

For each Meal

For each participant

Get eating occasion/create if required

For each food item in meal

Add copy of food item to eating occasion

End for

End for

End for

Human Interface Design

Overview of User Interfaces

Eat:

User clicks the EATING button.

If there is more than 1 participant, they must then select the Participant to record an eating occasion. Otherwise the one participant is automatically selected. And the user is progressed to the next page. Note, if the user clicks back, they will be returned to the select household member page even if they were automatically selected as the only household member.

The user can link recipes, this button will open a pop up which allows the user to check the recipes which are part of the eating occasion.

The user clicks the Add Food button which opens the camera. The user then takes an image and accepts the image.

The user then progresses to record the audio. If English version, they also have the option to record a text description instead. Once the user has accepted the text or audio they are returned to the Eating Occasion page which will now show the new Food Item and an Audio file which when clicked will play the recorded audio file, or if a text description was recorded a “Toast” pop up will appear containing the text. It is during the return from the recording audio to the Eating Occasion page the notifications to review the day and finalize the eating occasion are created and scheduled.

The user may long click the image to delete the food item. A confirmation pop up will appear to confirm the delete.

The user can record more food items or click the Submit button to return to the home page.

Shared Plate: If the version supports shared eating occasions the user may click “Meal” instead of “Own Plate” to capture a dish which is shared by multiple people (participants and non-participants). Shared dishes are created and handled the same way as the own plates however, the user will not be required to select the household member and will be directed to the eating occasion page.

Finalise Eat:

User clicks the Finalise Eating or, User clicks Notification to Finalise Eating Occasion.

If there is more than 1 participant, they must then select the participant to finalise eating occasions for. If the participant has an eating occasion to be finalised an Asterix (*) will be appended to their name. Otherwise the one participant is automatically selected and progressed to the next page.

The user is presented with a list of Eating Occasions which can be finalised. If there are many images, they can be horizontally scrolled to give a preview of the food items. To begin finalising the user must click the Finalise button.

For every food item in the eating occasion the user is presented with the image of the food item and three options: Ate All, Ate Most and Didn’t Eat.

Ate All: No need to take leftover image/audio so finalises the food item and the app moves on.
Ate Most: Moves to the camera to take an image of the leftovers and then to the audio to capture an audio description. Once audio is recorded the food item is finalised and the app moves on.
Didn't Eat: Finalises the food item but flags it as not eaten. The app then moves on.

Shared Plate: When the user clicks finalise the app creates a new eating occasion per household member and duplicates each (shared) food item from any shared plate 'meals' that have been recorded. These duplicated (shared) Food Items will contain a reference to the original food item they were copied from as a way of linking them all together. Each household member will have to finalise their own copy of the shared dish food items.

When a shared dish (duplicate) is first finalised by any participant, the user will be asked if there were any leftovers of the shared dish. If the user selects "Yes" they are moved to the camera followed by the audio recording screens to capture any information about the leftovers. If they select "No" they proceed to capture the Guest information.

A pop up will appear asking the user to enter how many (**including** those members participating in the study) people shared the dish. The user may use the + and – buttons to increase the counter for each category (Adult Male, Adult Female and Children). After leftover and guest information is captured once for a (shared) food item, the results are copied to all other duplicates of the (shared) food item, meaning no other participant is required to capture this information again and will be directed to the next step automatically.

Each participant will then have to answer if they ate or did not eat from the shared dish.

Cook:

User clicks the Cook button and is taken to the Recipes page.

A list of recipes is displayed with the final image of the recipe displayed first.

The user clicks the Red plus button to add a new Recipe. User is taken to the create recipe page.

User can click the microphone button to record the name of the recipe. If English version a text recipe name can be entered instead of audio file. Once recorded an audio file icon appears which when clicked will play the audio or show a "Toast" message containing the name.

Ingredients can be added by clicking the add ingredient button. The user is taken to the camera to take an image of the ingredient. Once the image is accepted, they are moved to the audio recording screen. No text descriptions are permitted for describing ingredients. Once the audio has been accepted the user is moved back to the Create Recipe screen. The image of the ingredient is now shown with an audio file icon which once clicked will play the audio description of the ingredient.

The user may long click the image to delete the ingredient. A confirmation pop up will appear to confirm the delete.

Once all the ingredients are captured the user can click the Submit button. A prompt to take an image of the final product will appear, if the user wishes to take the image now, they can click yes and will be moved to the camera. If they click no they will continue back to the create recipe page.

The recipe will now be displayed with the final image as the first image in the list. If no final image was taken a placeholder icon will be displayed instead.

If the recipe is unlocked the user can click the edit button to return to the create recipe page with the Ingredients loaded to add or delete ingredients. A recipe becomes locked when the record review process begins.

The user can long click the edit button to delete the recipe. A confirmation pop up will appear.

Breastfeed

If a household member has been created which is breastfed, the Breastfeed button will be displayed on the main screen. Once clicked the user is taken to the select household member page to select the participant being breastfed. If the user selects a participant not being breastfed an error “Toast” will be displayed. Once an eligible participant has been selected the app moves to the breastfeed page.

The user can press the breastfeed button to record the time at which the breastfeed occasion takes place. A breastfeed occasion is a regular eating occasion however it does not require finalising.

Record Review

Each recording day requires the participants to finalise their day. At the scheduled time (set when setting up the household) a notification will be sent to remind the user to review their day. This notification when clicked will take the user the record review screen. The notification will also be added to the list of notifications which can be accessed from the main screen by clicking the bell icon. These notifications will remain in the list for 24hrs or until they are “seen” whichever is greater. A notification from the bell menu becomes “seen” once the “Day Complete” button is clicked.

When the record review screen is first accessed all the recipes are locked so they can no longer be edited.

On the record review screen the user is shown the images of all the food items grouped by eating occasion. With the question if they ate anything else for the day.

The user may click yes to make an audio only record of what they ate but did not capture an image for. If English a text description may be made.

When the user clicks “Day Complete” all the eating occasions are finalised along with the food record. The app then moves on to the main home screen.

Verify

Only available through the settings menu behind the PIN number is the verify screen. The verify screen displays all the eating occasions, for all the food records for each household member. Each food record will either have a green title (time and day of food record) which symbolizes the food record has been reviewed via the end of day review use case (Record Review), or a red title which symbolizes the food record has not been reviewed. The user can click the title which will move to the Record Review page for that food record from which they may review and add any audio only records as required. Once "Day Complete" has been clicked the user will be returned to the Verify page and the previously red title will now be green.

Reminders:

Unfinalised Eating Occasion

Created: When user adds a food item to an eating occasion.
Delivery Time: 1 hour after the final food item is added to the Eating Occasion
Action: Opens screen to select eating occasion not finalized
Note: A notification will also appear under the bell icon for unfinalized eating occasions. This will clear once the eating occasion has been finalized.

Record Review

Created: When user adds a food item to an eating occasion.
Delivery Time: Time scheduled during household set up. Default 7:30pm.
Action: Opens screen Review day for given food record.
Note: A notification will also appear under the bell icon for the user to review their day. This will clear once the day has been reviewed (Clicked the "Day Complete" button) and 24 hours since the delivery of the notification has passed.

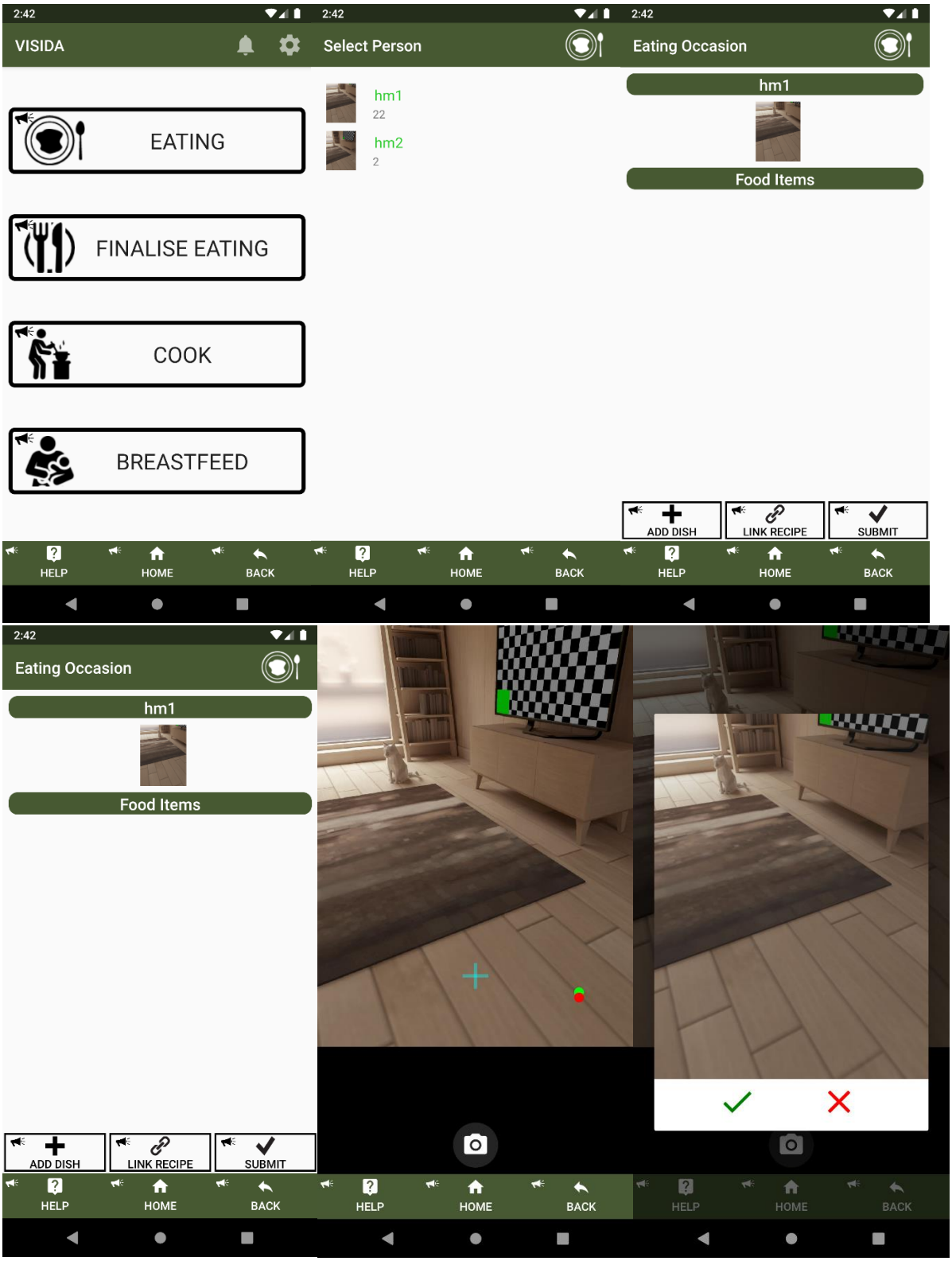
Reminder to Record for the Day

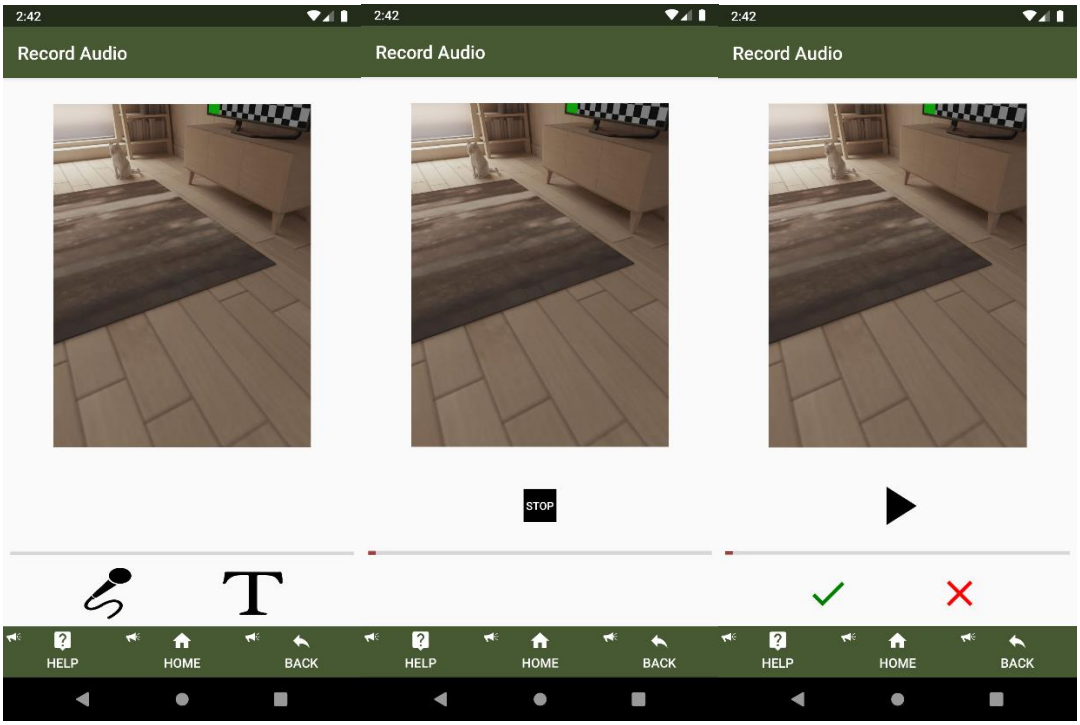
Created: When user sets times for reminders during household set up.
Delivery Time: At time/date scheduled.
Action: Opens the Application to the home screen.

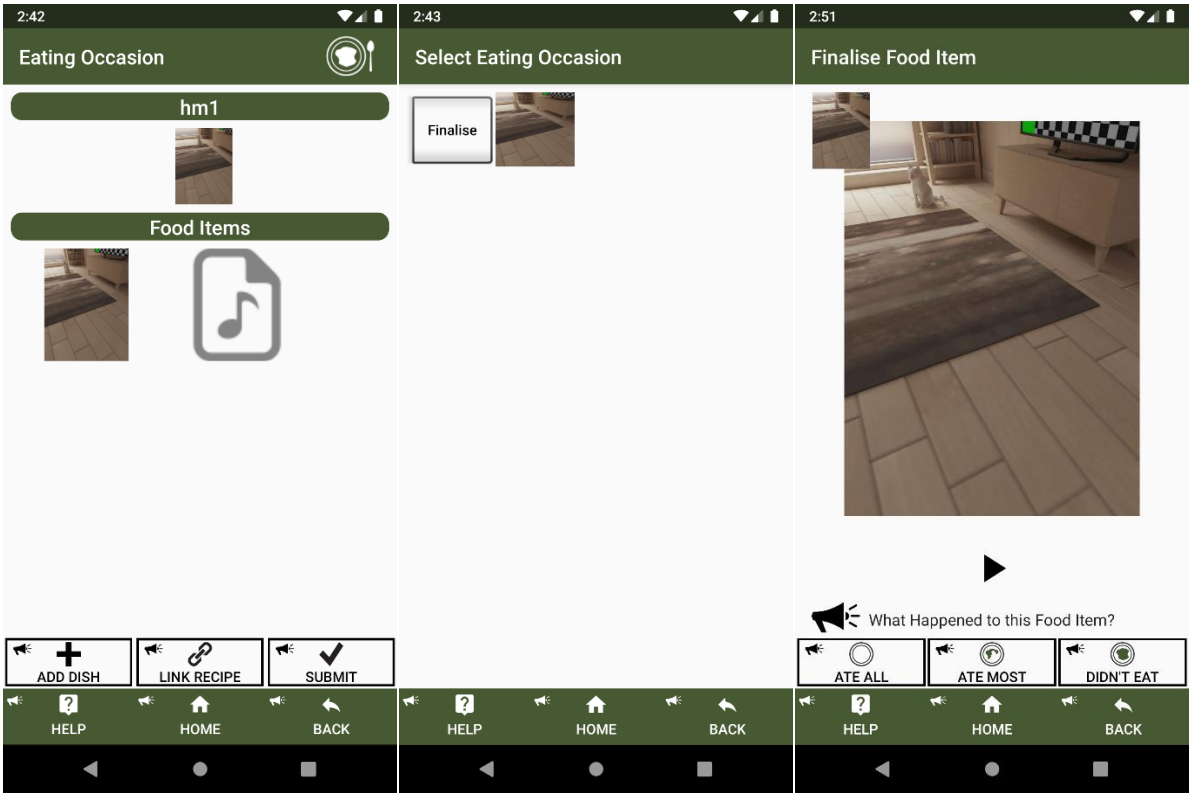
Reminder to wear sensor

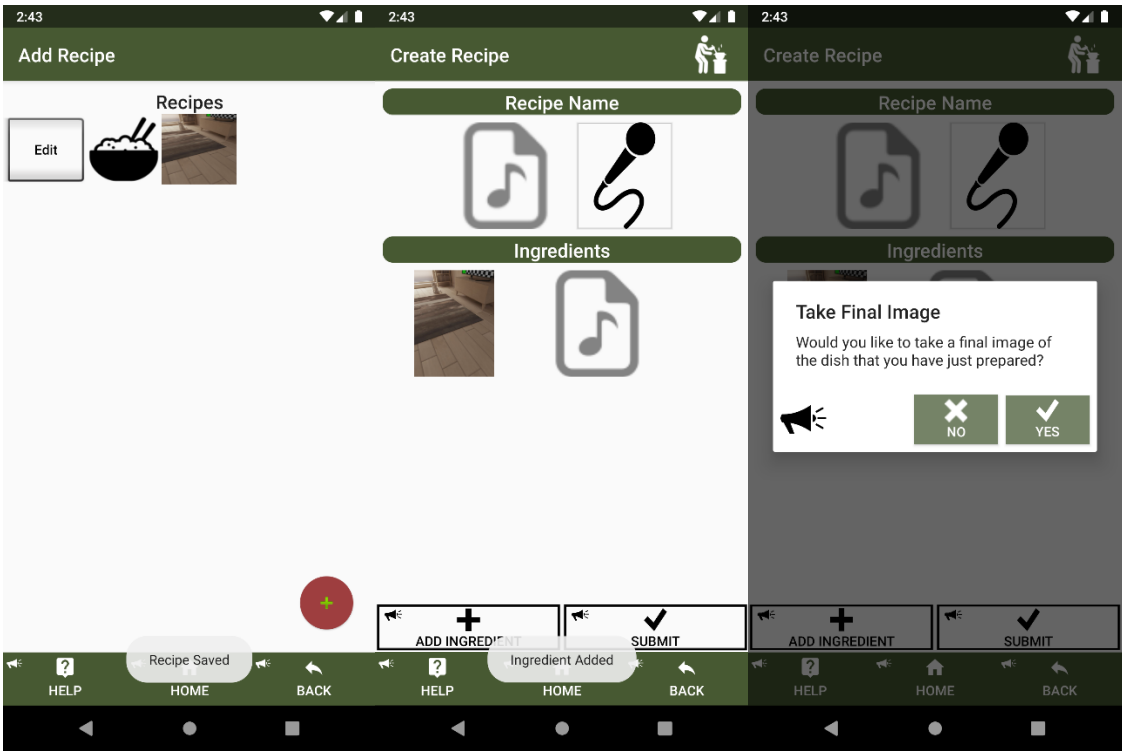
Created: When user sets times for reminders during household set up.
Delivery Time: At time scheduled on the days the Reminders to record are scheduled.
Action: Opens the application to the home screen.
Note: This notification contains an audio button which will play the instruction to wear the sensor.

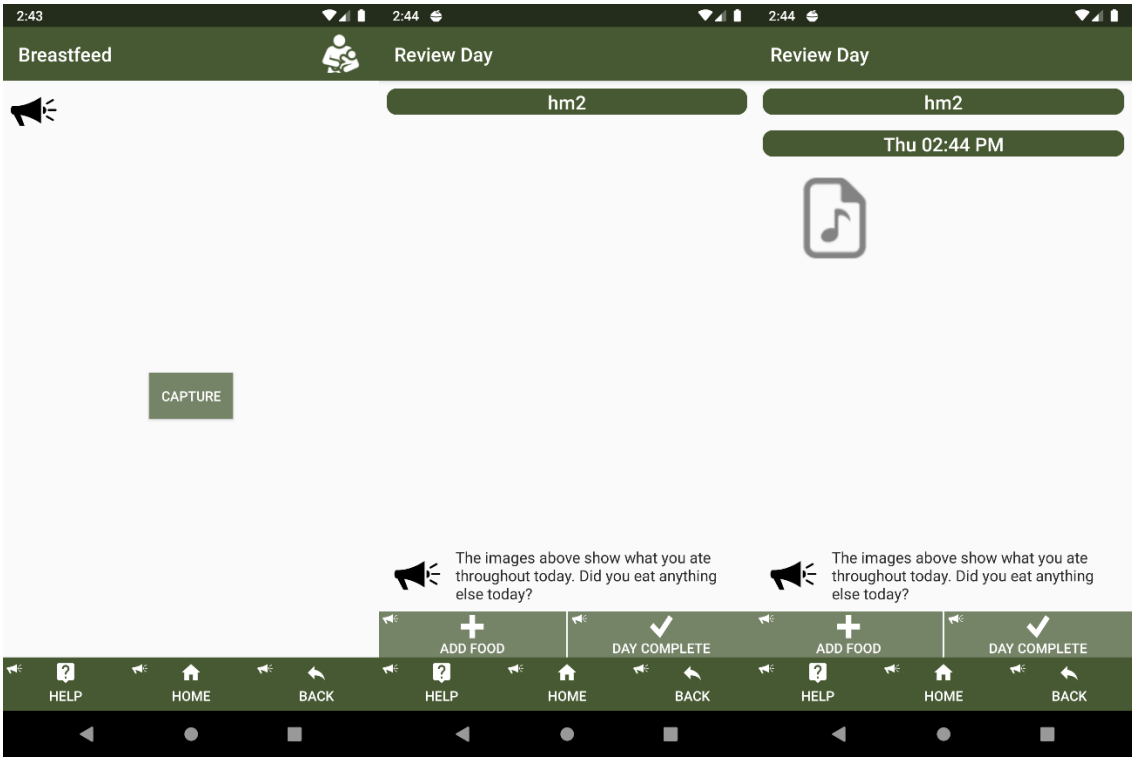
Screen Images




2:42
Record Audio
STOP
HELP
HOME
BACK


2:42
Eating Occasion
hm1
Food Items
ADD DISH
LINK RECIPE
SUBMIT
2:43
Select Eating Occasion
Finalise
2:51
Finalise Food Item
What Happened to this Food Item?
ATE ALL
ATE MOST
DIDN'T EAT
HELP
HOME
BACK


2:43
Add Recipe
Recipes
Edit
HELP
HOME
BACK
Recipe Saved
2:43
Create Recipe
Recipe Name
Ingredients
ADD INGREDIENT
SUBMIT
HELP
HOME
BACK
Ingredient Added
2:43
Create Recipe
Recipe Name
Ingredients
Take Final Image
Would you like to take a final image of the dish that you have just prepared?
NO
YES
ADD INGREDIENT
SUBMIT
HELP
HOME
BACK


2:43
Breastfeed
CAPTURE
HELP
HOME
BACK
2:44
Review Day
hm2
The images above show what you ate throughout today. Did you eat anything else today?
ADD FOOD
DAY COMPLETE
HELP
HOME
BACK
2:44
Review Day
hm2
Thu 02:44 PM
The images above show what you ate throughout today. Did you eat anything else today?
ADD FOOD
DAY COMPLETE
HELP
HOME
BACK

Appendices

Image/Audio Filenames

Each field is separated by an underscore. Where:

HouseholdID: The id given to the household during set up.

PPID: The participant ID given to the participant during household member creation

FRID: The Food Record Id

EOID: The eating occasion Id

FIID: The food item Id

MealId: The meal Id

Timestamp: The time the image was saved. In the format yyyy-MM-dd-HH-mm-ss

Image	File Name
Food Item	HouseholdID_PPID_FRID_EOID_FIID_TIMESTAMP.jpg
Audio File	HouseholdID_PPID_FRID_EOID_FIID_TIMESTAMP.mp3
Ingredient Image	HouseholdID _RecipeID_IngredientID_timestamp.jpg
Ingredient Audio	HouseholdID _RecipeID_IngredientID_timestamp.mp3
Shared Dish Image	HouseholdID _MealID_Timestamp.jpg
Shared Dish Audio	HouseholdID _MealID_Timestamp.mp3
Recipe Name Audio	RECIPE_HouseholdID _RecipeID_Timestamp.mp3
Audio Only Record Audio	HouseholdID_PPID_FRID_EOID _FIID_ TIMESTAMP_ AUDIOONLY.mp3

Development Pipeline

Test Driven Development

Write a test for the feature to be implemented. Tests are either unit tests: app/src/test/../visida or Android Tests: app/src/androidTest/../visida for Instrumentation tests run on the emulator. Run the test expecting it to fail.

Develop the feature/bug fix. Re-run the test and ensure the test passes. Repeat this process for each step of the development of the feature. Once completed the feature should be implemented and a number of unit tests should now exist to test regression.

When doing manual tests on the emulator, it can be useful to access the app's built in sqlit3 database. To access this database ensure the emulator image is rootable. Then in the terminal from within android studio

- Adb devices (will list the devices if more than one device you will need the device ID for each of the following ocmmands, if only one then it will automatically use the single device)

- Adb root
- Adb shell (when the shell prompt appears make sure it has the pound # symbol, this means you have root access, if not the adb root did not work, you may not have a rootable image).
- Cd to the VISIDA directory: cd data/data/au.edu.newcastle.jnc985.visida/databases
- Run sqlite3: sqlite3 appdatabase.db
- List the tables: .tables

Tools/Versions

Name	**Version**
Java	1.8 (Java 8)
Android Studio	3.4.1
JVM	OpenJDK 64-Bit Server by JetBrains
Gradle	5.1.1 (classpath 'com.android.tools.build:gradle:3.4.1')*
Library Dependencies	See /app/build.gradle

* Upgrading from Gradle 3.4.1 caused a build error with styled attributes so remaining on 3.4.1