

Project Title: Infralog

Team: Team Enigma

Authors: Blake Husserl, Rafal Wojciek, Jose Vasquez, Metehan Ozten, Ahmad Bayonis

Revision History:

- (1/26/15) Document created
- (1/28/15) First Draft finished, Intro, Glossary, System Architecture, Requirements
- (2/23/15) Updated Glossary, System Architecture Overview, and Use Cases.

Introduction:

The purpose of this document is to detail the requirements of the application, hereafter known as the Infrared Wall Chart Application. Specifically the Infrared Wall Chart Application is a mobile application that will have two primary purposes. The first function is to serve as an educational or reference tool. It will have the ability to present users with the different figures and graphs that are present on the Infrared Wall Chart. The second function of the application is similar to that of a catalogue. It will help users search through a selection of Raytheon products using a selected range on the electromagnetic spectrum.

Glossary of Terms:

- Infrared Wall Chart - a chart developed by Raytheon that contains many helpful tools for working with and developing technologies that span the electromagnetic spectrum.
- RFDP - Raytheon file distribution protocol
- EM - Electromagnetic.
- RVS (Raytheon Vision Systems) Database - The database that holds the files that users can search for and download, this will be implemented via amazon RDS.

System Architecture Overview:

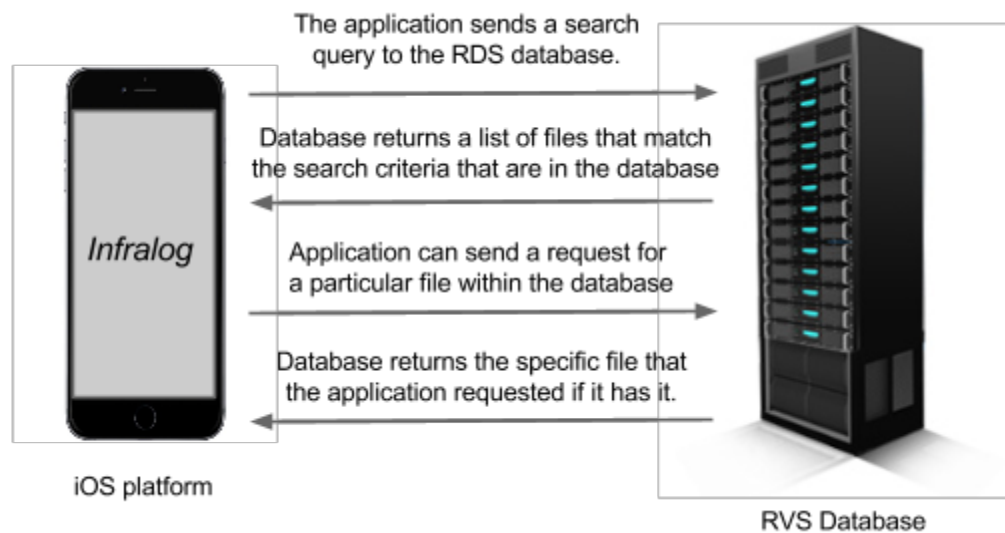


Figure 1: Graphic that shows the interactions between the system components

Requirements:

Use Case: Loading Welcome Screen

Actors: User

Precondition: The user has just finished logging in. This would be the front/welcome screen

Postcondition: The application is loaded successfully and responsive to user interactions

Flow of Events:

1. The top half of the screen is a button that says basic search.
2. The bottom half of the screen is a button that says advanced search.

Exceptional Cases: If the app does not load or is non-responsive it should produce an error, shutdown the application, and restart it.

Use Case: Transition from welcome screen to EM Spectrum Screen

Actors: User, Application

Precondition: The user is on the welcome screen and has selected the basic search button.

Postcondition: The user is on the EM Spectrum Screen

Flow of Events:

1. The system has registered that the user has selected the basic search button.
2. It takes the user to a new screen.
3. This screen has a picture of the EM spectrum which the user uses to specify a range of wavelengths.

Exceptional Cases: If for some reason it fails to transition/load the EM screen it should remain on the welcome screen and produce an error.

Use Case: Submit Search Query, EM Spectrum Screen (application perspective)

Actors: User, Database

Precondition: The user is at the EM spectrum screen and specified a specific region of the EM Spectrum.

Postcondition: Sent search successfully and waiting for results.

Flow of Events:

1. The application sends the search criteria to the database, and requests results
2. A new window with the search results opens up

Exceptional Cases: If there is an error with submitting the search or returning the results an appropriate error will be shown.

Use Case: Selecting a Range in the EM Screen

Actors: User, Database

Precondition: The user is in the EM screen, and has been asked to specify a range.

Postcondition: The sliders have been moved to the areas that the user desires.

Flow of Events:

1. By dragging the two sliders, the user can specify a range on the EM spectrum they're interested in.

2. When done adjusting the range the user can select the done button to be taken to a list of matched products.

Exceptional Cases: If the sliders are not responding the application should produce an error message and restart the application.

Use Case: Selecting a Range in the EM Screen (Via pre-created ranges)

Actors: User, Database

Precondition: The user is in the EM screen, and has been asked to specify a range.

Postcondition: The slider has been adjusted to the region selected.

Flow of Events:

1. User is on EM screen
2. They select one of the pre-created regions
3. When done selecting the region the sliders on the EM bar will adjust accordingly.

Exceptional Cases: If unable to process it should produce an error and restart the application

Use Case: Changing range of slider values with multiple touches

Actors: User, SliderUI

Precondition: The user is in the EM screen, slider thumbs are set to default value (0-100%)

Postcondition: The slider thumbs adjusted by the user simultaneously

Flow of Events:

1. User is on EM screen
2. They touch the two slider thumbs with two fingers at the same time
3. The user simultaneously moves the two thumb indicators to bound a desired region on the EM spectrum
4. Application continuously reads the values of both thumbs to return when 'Search' is hit

Exceptional Cases: If unable to process it should produce an error and restart the application

Use Case: Receiving Slider Results (Application view)

Actors: User, Application, Database system

Precondition: The user has specified a frequency or range of frequencies they are interested in and submitted.

Postcondition: The results are on the display for the user to see

Flow of Events:

1. The request will be sent to the database
2. After processing the user will be redirected to a new window where the results of the query will be displayed.

Exceptional Cases: If the results are unable to be received then it should display an error message and then ask for them to submit the search again.

Use Case: Query has been received by database (database perspective)

Actors: Database

Precondition: User has submitted a search query

Postcondition: Database has pulled up all the files that match the search.

Flow of Events:

1. Database reads in the search criteria.
2. Database iterates over its files and pulls up all the ones that fit the criteria.

Exceptional Cases: If database is unable to do this it will send an appropriate error to the application.

Use Case: Sorting Results from User EM Spectrum Range Query (Database Perspective)

Actors: Database

Precondition: The database has pulled up all the files in the user's query.

Postcondition: The results are now sorted in the way that the user specified and ready to be sent back to user.

Flow of Events:

1. The database has pulled all the files that pertain to the query and will now sort them by the criteria specified.

Exceptional Cases: If it is unable to sort how the user requested it will just show the unsorted results and display a message similar to "unable to sort as requested".

Use Case: Log In/Register (Stretch Goal)

Actors: User, Database

Precondition: Application has been opened

Postcondition: User will have successfully logged in and be moved to the

Flow of Events:

1. Log in will only let authorized users use the application
2. If the user has a registered account, log in as usual. Otherwise register a new account.

Exceptional Cases: If the system is unable to login it will show an error message asking the user to log in again.

Use Case: Registering a new User (Stretch Goal)

Actors: User, Application, Database

Precondition: User has needs to register to use the application

Postcondition: User is now successfully registered and logged in.

Flow of Events:

1. User is asked to submit an email and a password.
2. The application checks with the database to see if it is valid.
3. After verifying it allows the User to log-in.

Exceptional Cases: If an email or password is invalid it will let the person know accordingly, or if it is unable to connect to the database it will let the user know accordingly.

Use Case: Save login information (Stretch Goal)

Actors: User, application

Precondition: User is in the login page, with login information entered

Postcondition: User will have successfully logged in with 'save login' checked

Flow of Events:

1. The user checks the 'save login info' box
2. If the user has a registered account, log in as usual.
3. encrypt user password
4. save information on the device

Exceptional Cases: Application is unable to login due to network error or no existing account it will show the relevant error message and go back to the login page or registration page.

Use Case: Change email or password (Stretch Goal)

Actors: User, application

Precondition: User at the login page, and cannot remember login information

Postcondition: User will be redirected to a reset password or email page

Flow of Events:

1. The user clicks on 'Forgot password' button
2. App either shows a reset page, or redirects to relevant webpage

Exceptional Cases: Application is unable to connect to servers due to network problem. Show appropriate error and go back to login page.

Use Case: Logout (Stretch Goal)

Actors: User, application

Precondition: User is in any page, and intends to logout

Postcondition: User will have successfully logged out and is back to the main screen

Flow of Events:

1. The user checks clicks on the menu button
2. If the user chooses logout option
3. application logs out the user, and possibly forget login info
4. application switched back to login page

Exceptional Cases: If disconnected from network before logging out. Either save login with cookies, or automatically log out user.

Use Case: Selecting Technology specific PDF (user perspective)

Actors: User, Application

Precondition: The user is on the screen with the results from his recent query.

Postcondition: User is has downloaded a PDF that they selected

Flow of Events:

1. Specific Technology is selected from the sorted result of last query
2. Begin downloading the PDF from database.
3. PDF is downloaded to phone.

Exceptional Cases: If the PDF is not available an appropriate message will be displayed

Use Case: RSA Key Exchange (Stretch Goal) (system perspective)

Actors: Application, File Database

Precondition: PDF has been selected to download

Postcondition: System has successfully and securely sent the requested data

Flow of Events:

1. Application encrypts key
2. Application sends key to File Database
3. File Database decrypts and verifies key
4. File Database sends back desired PDF

Exceptional Cases: If any errors occur while attempting the RSA key exchange, the system will display an appropriate message error message.

Use Case: Mobile App Startup

Actors: User, Application

Precondition: The user is on the local mobile OS and is able to find the Raytheon Infrared Wall Chart App.

Postcondition: Application is loaded and the user is prompted with login

Flow of Events:

1. User selects application icon
2. Application is launched
3. Application takes control of input and output devices of platform.

Exceptional Cases: If the application is unable to load the system should prompt the user with an “Unable to load application” message.

Use Case: Open Reference Chart List

Actors: User, Application

Precondition: On any screen in application, not including login screen.

Postcondition: Have pulled open the list of reference charts available.

Flow of Events:

1. User presses the reference chart button (located in top right of screen).
2. From the right of the screen comes a list of all of the reference charts, which a user can select.

Exceptional Cases: If it is not working it will produce an error and restart the application.

Use Case: Reference chart selection

Actors: User, Application

Precondition: The user is at the reference chart page.

Postcondition: The reference chart is open on the screen

Flow of Events:

1. Selecting a button with the chart’s name will open up that specific chart in a new window (all of these charts are taken from the Infrared Wall Chart).

Exceptional Cases: If the system is unable to open the reference chart it will display a message saying it is unable to do so.

Use Case: Transition to the Advanced Search Screen

Actors: User, Application

Precondition: The user is on the welcome screen.

Postcondition: The user is on the advanced search screen.

Flow of Events:

1. The user selects the Advanced Search on the welcome screen.
2. Application transitions to the advanced search window

Exceptional Cases: If the transition cannot be processed it will show an error message.

Use Case: Advanced search submission.

Actors: User, Application

Precondition: The user is in an advanced search screen to find specific information

Postcondition: Results of the advanced search are displayed

Flow of Events:

1. The user is allowed to use drop menus or something similar to specify certain categories to search in
2. There is a field to add the user's own keyword for the search

Exceptional Cases: If the advanced search is unable to be performed the system will notify the user and ask them to try changing their parameters.

Use Case: Advanced search by typing

Actors: User, Application

Precondition: The user is in the advanced search screen and clicked on the search bar.

Postcondition: A product(s)' name is displayed on the screen.

Flow of Events:

1. The user types into the search bar.
2. As the user types, a list of products matching the given input are displayed.
3. The search ends when the user is no longer typing and has found a product(s).

Exceptional Cases: If no products match the given input, no product names are displayed.

Use Case: Advanced search retrieval.

Actors: User, Application, Database

Precondition: The user searched for a product and its name is displayed.

Postcondition: The user can see the product's documents.

Flow of Events:

1. The user clicks on the name of the product.
2. The application sends retrieves the product's documents from the database.
3. The documents are displayed to the user.

Exceptional Cases: If the product's documents are not found on the database, an error message will be displayed.

Use Case: Application is suspended

Actors: User, Application

Precondition: The user is using the application.

Postcondition: The user is no longer using the application, but it remains open.

Flow of Events:

1. The application is suspended when the user locks their phone, clicks the home button, or switches to another application.
2. The application needs to save the user's current location and any information given.

Exceptional Cases: If an error occurs, the app should be restarted.

Use Case: Selection of Raytheon contact information

Actors: User, Application

Precondition: The user is in the main page

Postcondition: The user is on the page with all the contact information.

Flow of Events:

1. The user selects the "Contact Info" button
2. page shifts to page with contact Information

Exceptional Cases: Close the application.

Use Case: Contact Raytheon by phone

Actors: User, Application, Phone OS

Precondition: The user is in the Contact information page

Postcondition: The phone initiated a call with the chosen contact

Flow of Events:

1. The user selects the a contact from the list by clicking on it
2. The app sends the number to the OS phone app
3. A call is initiated to the specified number

Exceptional Cases: A problem in interacting with the OS phone app, could either cancel operation and return to contacts page, or have a copy number option

Use Case: Contact Raytheon by email

Actors: User, Application, Phone OS

Precondition: The user is in the Contact information page

Postcondition: The phone started a new email message with the chosen contact

Flow of Events:

1. The user selects the a contact from the list by clicking on it
2. The app sends the email address to the phone mail application
3. A new message is opened to the selected address
4. Could include information about product in question in the email

Exceptional Cases: A problem in interacting with the OS mail app, could either cancel operation and return to contacts page, or have an option to copy the email address

Use Case: PDF not received from server

Actors: Application

Precondition: Query sent to server

Postcondition: display “file not found”

Flow of Events:

1. File Request/Response Cycle ends unsuccessfully
2. Application ceases to try to retrieve PDF
3. Pop-up of text saying “File not found”

Exceptional Cases: Close the application.

Use Case: Transition between pages failed

Actors: Application

Precondition: The user has selected an action that would lead the application to transition to another layout

Postcondition: Transition to homepage

Flow of Events:

1. Transition fails
2. the Application realizes it is using an invalid layout
3. Application transitions to homepage
4. Error pop-up appears

Exceptional Cases: Close the application.

Use Case: File-Index Request/Response Cycle

Actors: User, Application, Server

Precondition: The user is using the application and sends a search request from the application.

Postcondition: The user has submitted the search and the server has responded with the results.

Flow of Events:

1. User opens up the application and selects either “basic search” or “advanced search”
2. Once the search page is up, the user fills in the relative fields and submits the search
3. This will open up a TCP connection with our Raytheon File Distribution Protocol server, which will then convert the selected filters into a mysql ‘SELECT’ statement, and then return the list of matching(relevant) products.
4. Once the user receives the list of products, it deserializes the packets and presents them in list format to the user.

Exceptional Cases: If TCP connection isn’t opening or, the server isn’t responding, retry the connection 3 times, then if it still isn’t working print error and close the application.

Use Case: File Request/Response Cycle

Actors: User, Application, Server

Precondition: The user is using the application and sends a file request from the application.

Postcondition: The user has submitted the search and the server has responded with the necessary files

Flow of Events:

1. User opens up the application and performs a search and receives a list of files
2. The user selects a product he is interested in and the client will send a file request to the RFDSP server.
3. The RFDSP server grabs the file and then sends it across the TCP connection.
4. The client receives the bytes across the wire, saves the file and then opens it up for the client

Exceptional Cases: If TCP connection isn't opening or, the server isn't responding, retry the connection 3 times, then if it still isn't working print error and close the application.

Use Case: Application is closed.

Actors: User, Application

Precondition: The user is using the application.

Postcondition: The user is no longer using the application and it has been closed.

Flow of Events:

1. The application is terminated when the user closes it by turning their phone off or swiping the application up when the user double clicks the home button.
2. The application is terminated without a need to save user location.

Exceptional Cases: Close the application.

Use Case: Button fails to respond/perform function

Actors: Application, User

Precondition: User has pressed button.

Postcondition: Page screen has been restarted.

Flow of Events:

1. User presses button
2. Button fails to respond
3. Application reloads the page

Exceptional Cases: If the page fails to reload the application will exit and restart.

Use Case: Translating search criteria to protocol

Actors: Application, User

Precondition: Search Criteria has been specified by user

Postcondition: Successfully translated to bits specified by the protocol

Flow of Events:

1. Application takes the user's criteria
2. Follows the protocol specified and translates it to appropriate bits (protocol can be found on the github).
3. The bits will be sent to the server.

Exceptional Cases: If the application fails to translate the bits it will restart the page for the user to respecify criteria.

Prototyping Code and Test Cases:

<https://github.com/BlakeHusserl/RaytheonCapstone>

System Models

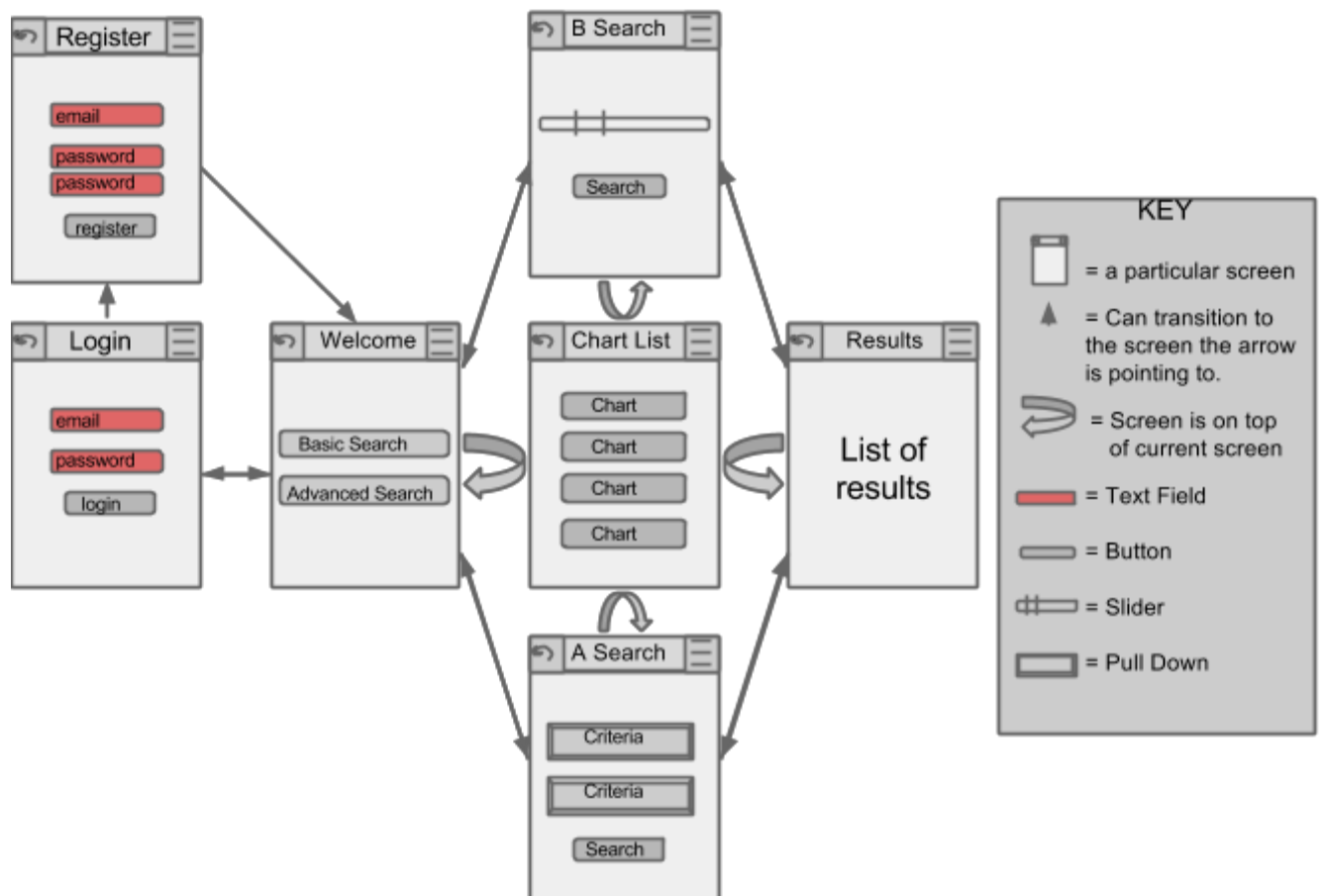
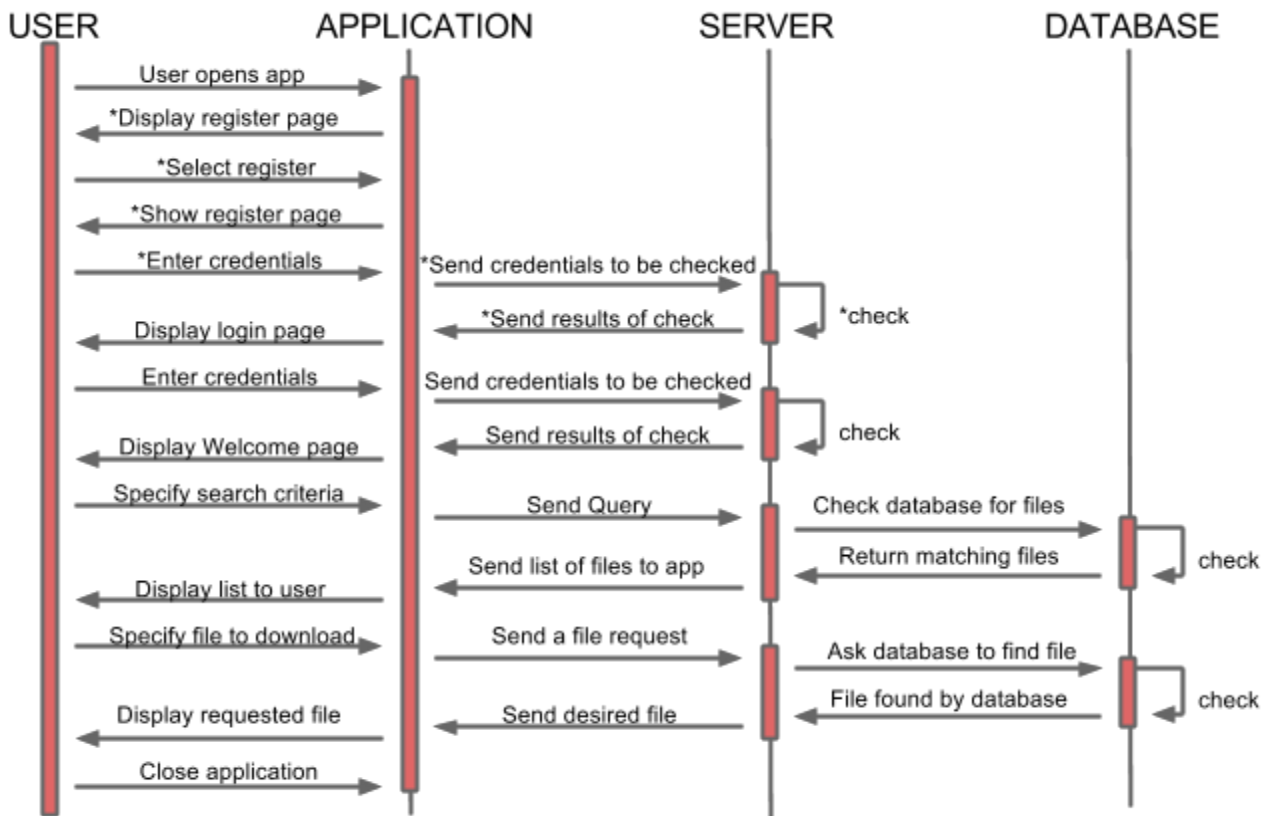


Figure 2: The Screen Layouts and Transitions of the Application



Note: * means that it is for First time user only

Figure 3: Sequencing diagram of the application/database operations

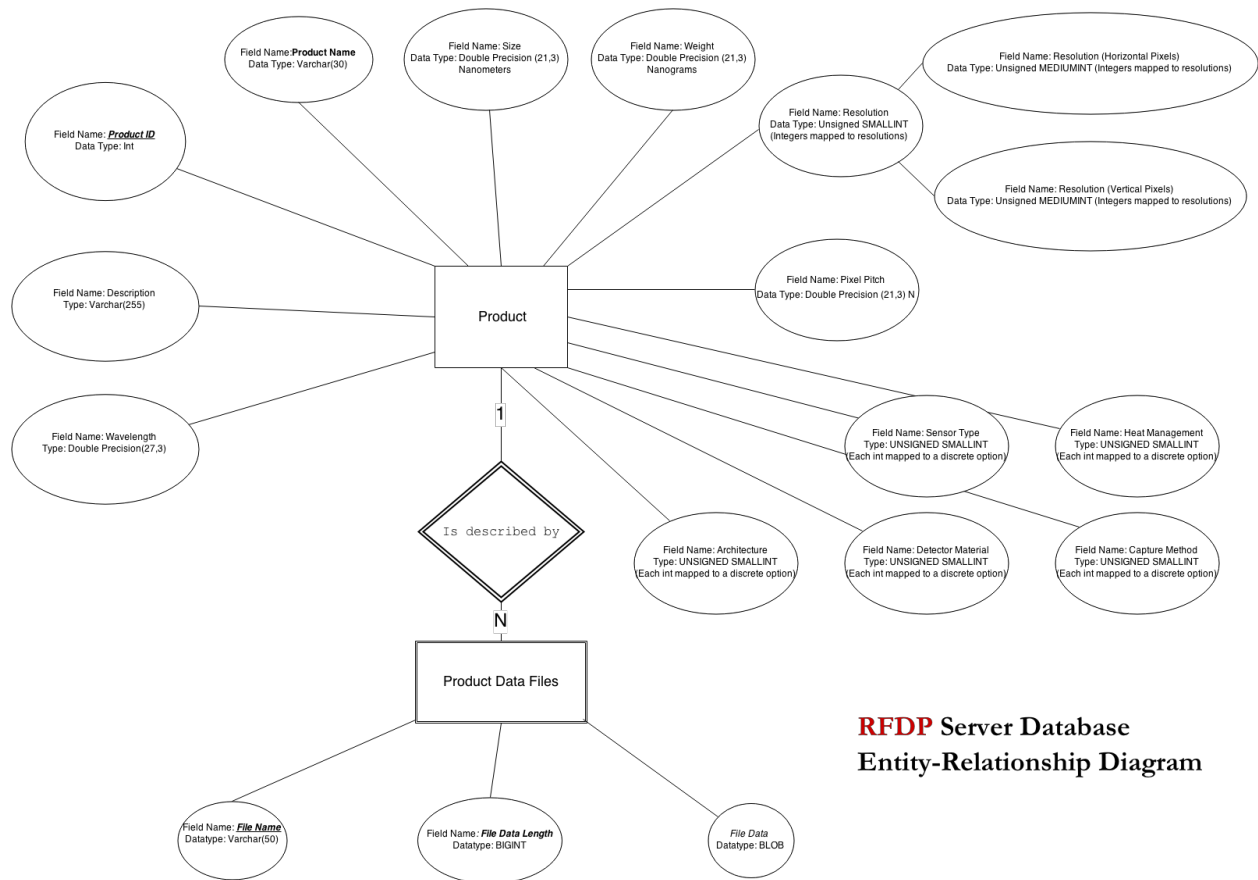


Figure 4: Server-Side Database ER Diagram, see link below for better view of the diagram
<https://drive.google.com/file/d/0B7rAlt4QMCwOWmRSSHVJN2JDUDA/view?usp=sharing>

Appendices:

Technologies employed:

- Amazon RDS - used for the server/database platform
- Xcode and Objective C - used for coding the iOS application
- iOS - chosen application platform
- MySQL - used to develop the database