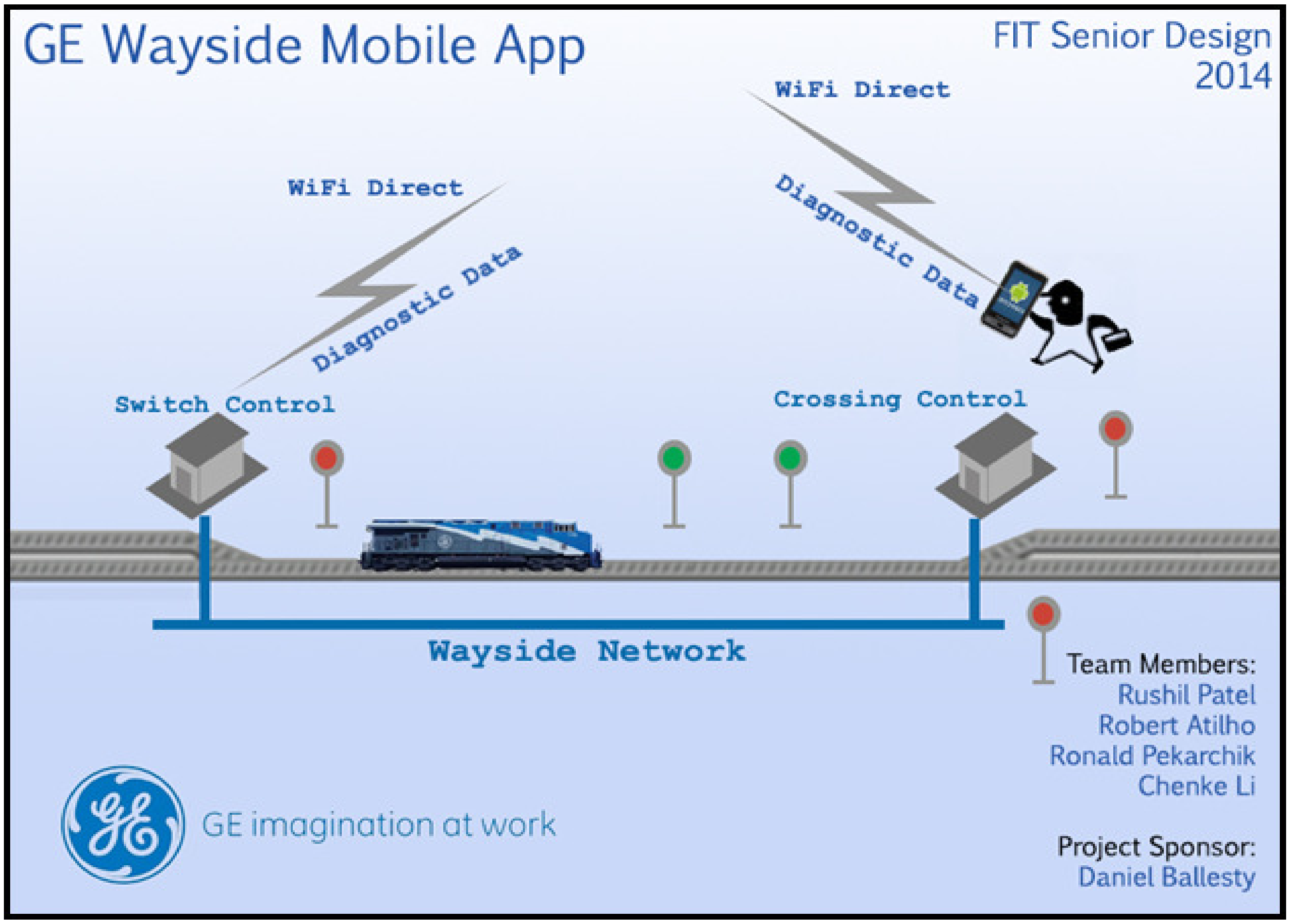


Milestone 1 - Design Document

Team Members:

Rushil Patel	rushil2011@my.fit.edu
Robert Atilho	ratilho2012@my.fit.edu
Ronald Pekarchik	rpekarch2006@my.fit.edu
Chenke Li	lic2012@my.fit.edu

Faculty Sponsor:

Daniel Ballesty (GE)	Daniel.Ballesty@ge.com

CS Faculty Sponsor:

Dr. Liam Mayron	lmayron@fit.edu

Application Layout

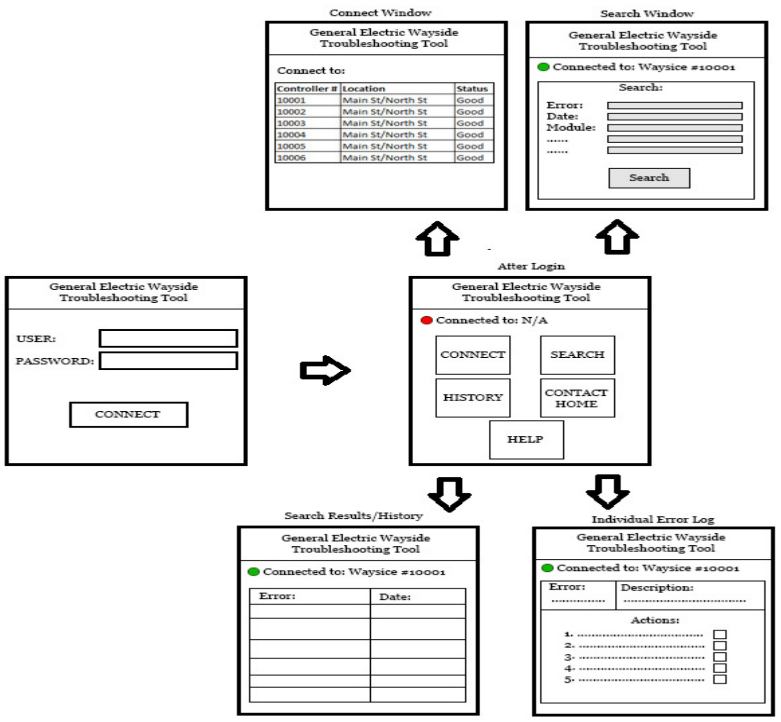
The Android application will be broken up into roughly six screens for the user. The first screen will ask the user to enter their username and password. In order to move on to the next screen, the user's credentials must be verified. If the user cannot be verified, the options for an email to reset their password, create a username, or retrieve a forgotten username will be shown.

Following the login screen, the main screen will consist of several buttons to do different tasks in order to perform the tasks related to the wayside controller. Connect to the box will be the first thing the user must do in order to get the latest logs from the wayside controller. After this has been done, the user will have the choice of searching logs, looking at log history, contacting General Electric, or help with the application.

On the search screen, the user will have several field options that can be filled out in order to get different search results of the logs. This will consist of time/date, status of track, conditions of track, module that printed the log, etc.

Looking at log history will just consist of a general overview of all the logs that have been downloaded from the wayside controller so that the user can just scroll through and look at everything. This screen will obviously have limited data due to the size of the device's screen. Flipping the phone to it's side will allow the user slightly more details for the headers.

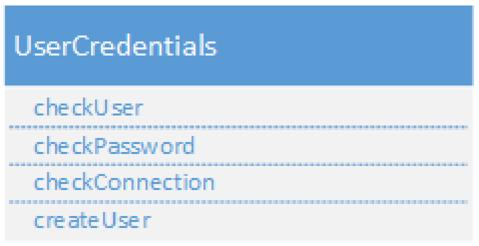
The contact screen will consist of several choices. These being, to send an email to the technicians back at General Electric, call the IT department, etc. The help button will offer any tips or tricks for the user to better understand the different aspects of the application and to provide guidance when the user cannot accomplish a specific task such as connecting or searching the logs.


Connect Window
General Electric Wayside Troubleshooting Tool
Connect to:
Search Window
General Electric Wayside Troubleshooting Tool
Connected to: Waysice #10001
Search:
Error:
Date:
Module:
......
......
Search
After Login
General Electric Wayside Troubleshooting Tool
Connected to: N/A
CONNECT
SEARCH
HISTORY
CONTACT HOME
HELP
General Electric Wayside Troubleshooting Tool
USER:
PASSWORD:
CONNECT
Search Results/History
General Electric Wayside Troubleshooting Tool
Connected to: Waysice #10001
Error:
Date:
Individual Error Log
General Electric Wayside Troubleshooting Tool
Connected to: Waysice #10001
Error:
Description:
Actions:
1.
2.
3.
4.
5.

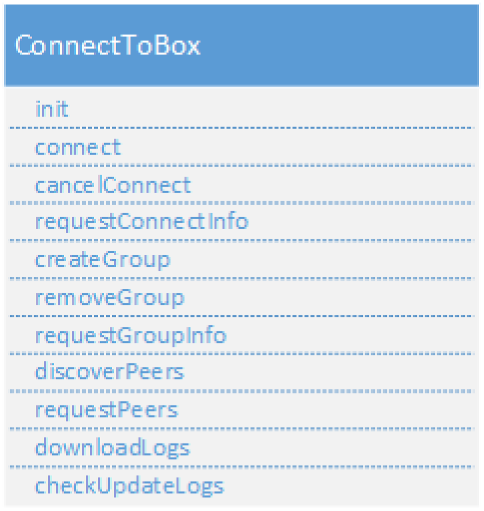
Controller #	Location	Status
10001	Main St/North St	Good
10002	Main St/North St	Good
10003	Main St/North St	Good
10004	Main St/North St	Good
10005	Main St/North St	Good
10006	Main St/North St	Good

UML Diagram of Classes & Methods

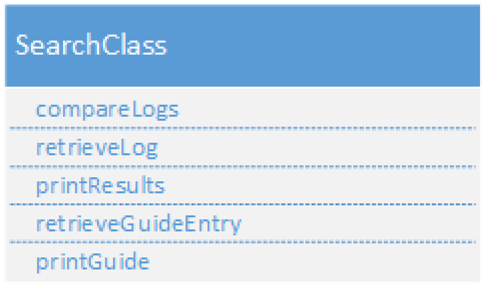
The UserCredentials class will be the first part of the application since the user must have access to be able to use the program. This part of the application must check to make sure that the user is able to connect to the database to check the user's credentials. If the user is not registered, the user must request to create a userID which then must be verified by General Electric. The user will not be able to use the application if the credentials cannot be verified.



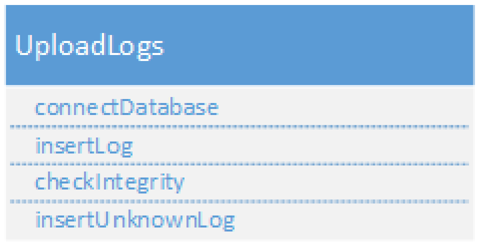
The most significant and largest class is the ConnectToBox class. This class will handle setting up the WiFi Direct connection which is used to connect to the wayside controller. The class uses an API from Android 4.x that will be built upon in our application. The downloadLogs function will gather all the logs from the wayside controller, while the checkUpdateLogs will compare those from previously downloaded logs to make sure the user has access to only the most current logs. If the user shall need access to previous records of the box, another method may be implemented, but for now all of that data shall only be obtainable when connected to a cellular network via the database storing all previous logs.



The SearchClass will handle all of the users requests in relation to log events that have been downloaded to the phone after the connection has been verified by the ConnectToBox class. This class will essentially have it's own window in order for the user to have ease of use when dealing with multiple fields. This class will also handle all of the printing to the screen of the guide book that the user must have access to in order to service the box and diagnose any problems that are occurring.

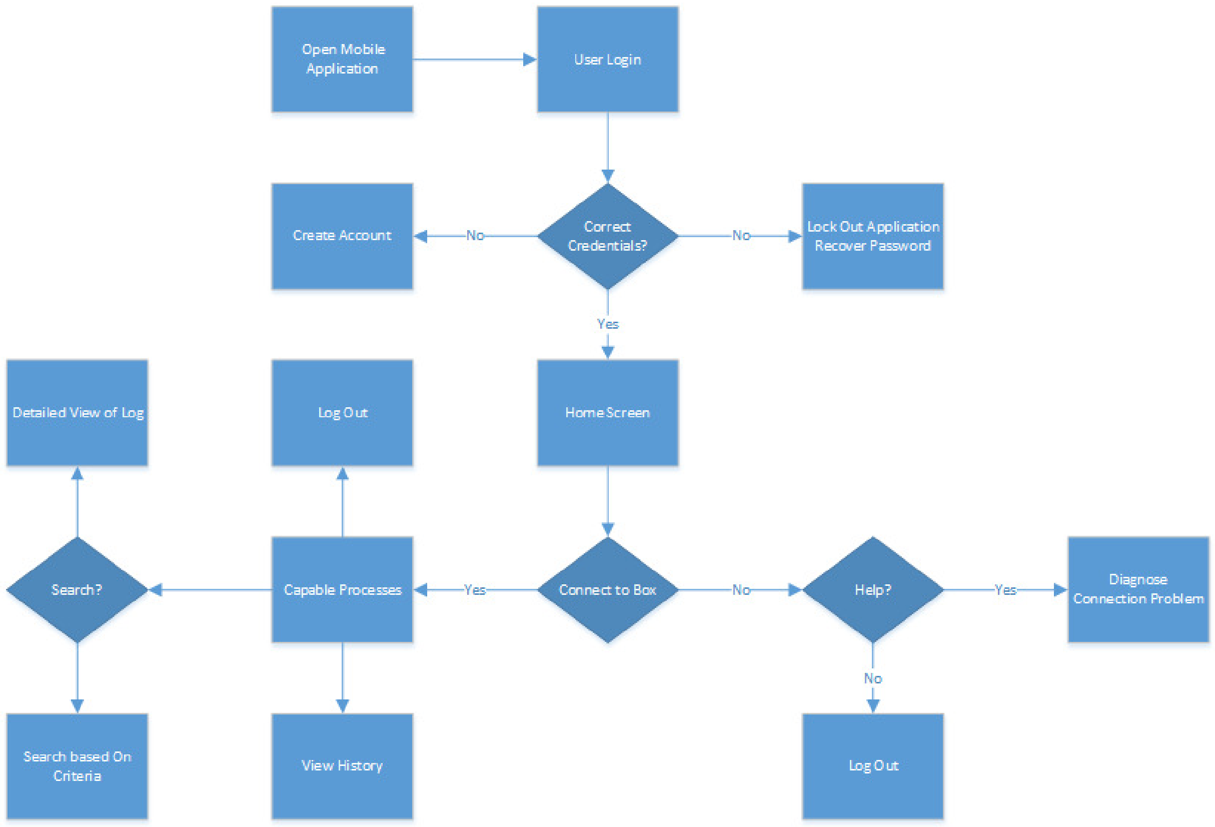


The UploadLogs class will handle sending new logs to the database on the server side via cellular network connection. This database will basically hold all of the logs that have been downloaded from all the wayside controller boxes in order to generate statistics in the future.



Flow Chart of Processes/Functions

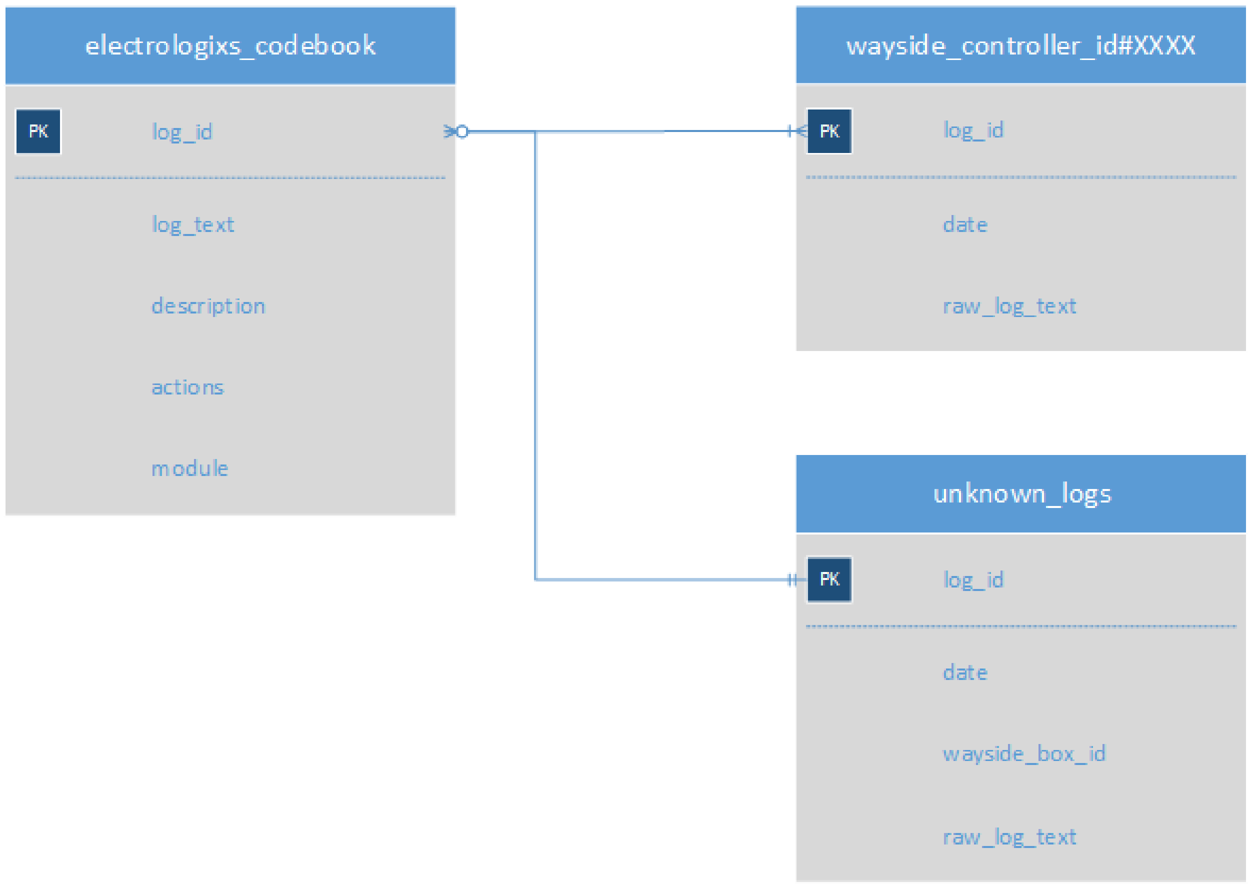
The application's flow chart is similar to that of the Application layout. The application will start off with a login in page, determining whether the user has permission to use it. There are three possible scenarios that can occur here. When the user verifies their credentials, the application shall move to the home screen and determine whether or not connection to the wayside controller is possible. Once this is determined the user will have either the option to get help if the connection was not established, or be able to do the other processes which include logging out, searching, viewing history, and obtaining a detailed view of logs.



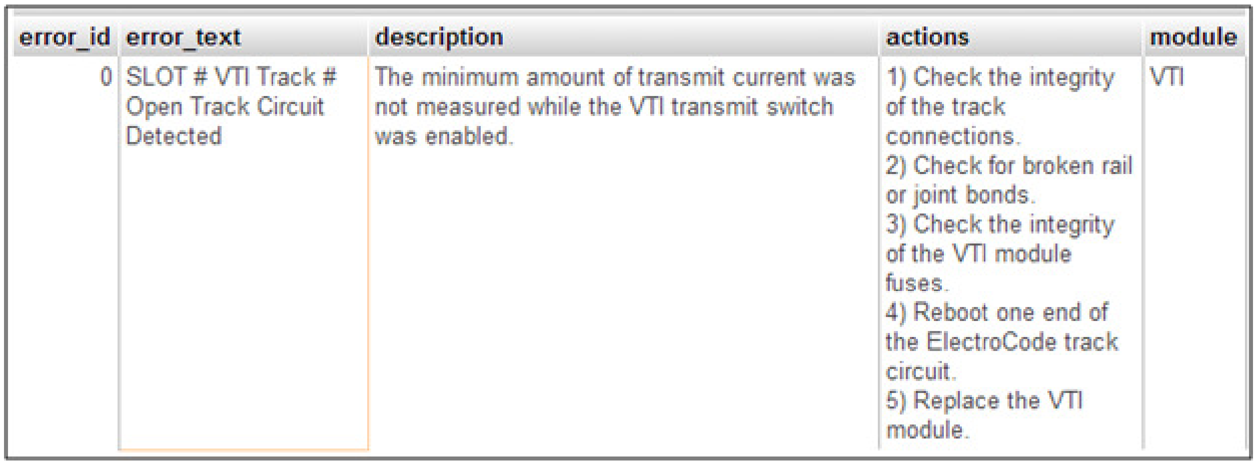
Database Structure

At the moment, there are potentially two databases. One will be located locally on the mobile device and another will be located on a central server that contains logging information for all of the boxes.

Mobile Device Database



Example Entry into electrologixs_codebook:



error_id	error_text	description	actions	module
0	SLOT # VTI Track # Open Track Circuit Detected	The minimum amount of transmit current was not measured while the VTI transmit switch was enabled.	1) Check the integrity of the track connections. 2) Check for broken rail or joint bonds. 3) Check the integrity of the VTI module fuses. 4) Reboot one end of the ElectroCode track circuit. 5) Replace the VTI module.	VTI

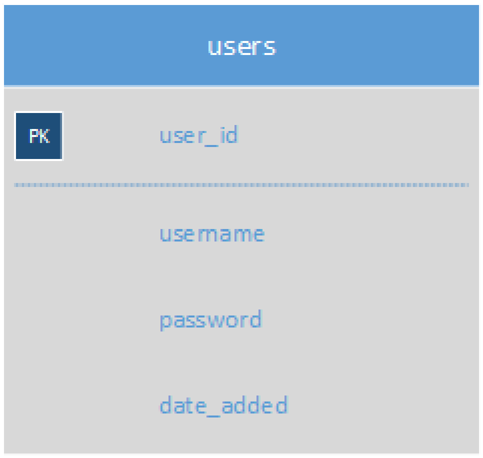
Example of raw data to be inputed into wayside_controller_id#XXXX:
02-11-14 11:52:44 B:TICGMAIN 20198 SLOT 1 VTI Track 1 Open Track Circuit Detected

In this example, our application will match this log entry to the entry in the codebook and set the error_id to that of the one in the codebook.

The unknown_logs is a table used to sort all of the unknown lines of data in the log file. This essentially means that there is no match for the log entry in the codebook. There are 75% of all logs recorded in the codebook. Whenever, a new unknown entry is encountered it will also be stored into the unknown_logs table for later analysis. Descriptions and actions for these unknown entries will be written up and stored into the codebook as a way of trying to obtain a more complete guidebook.

<u>Server Database</u>

The database on the server will essentially be the same structure as the one on the mobile device. The only added table would be the following for user verification purposes:


users
PK
user_id
username
password
date_added