Nathan Donaldson
U0632803
1/18/18

General Expert & Layperson Essay (Pro400)

The Pro400 is a device owned by Orbcomm that is installed into vehicles that require (by law) the driver's hours to be recorded in that vehicle. These laws vary in some states, but most of them are the same. These devices help companies track and maintain their company trucking fleets. Not only does the Pro400 keep track of HOS (hours of service), but it has focus in safety as well. Drivers have their hours tracked for many different reasons. One, for instance, is to keep track of your employees and how much they are working; two, is to reduce unsafe driving, which would mostly be sleep deprived drivers; and three, to keep the drivers sane, requiring them to take breaks before continuing to work. On top of that, there are the other safety features on the device. The Pro400's safety features keep track of things like seatbelt, ignition, movement, crash detection, location, battery, and many other things that a vehicle computer can output. Audio helps the drivers along the way as well, giving warnings when the driver may be speeding or turning to harshly, all to make sure the driver is safe. It also may prevent them from getting a ticket from the authorities as well. The Pro400 is installed in the windshield of a truck and is about the size of a wallet. It is oriented at about a 45 degree angle, giving it much better cell connection. A mobile application called Connect allows drivers to login and out of their devices, and look at a few informational tabs that help them verify some things. A new system has started called ELD(Electronic Logging Device) that has changed a few things.

For those of you who are familiar with our devices, the new ELD mandate has required new features that started this year. Luckily if you had an HOS device previously before this mandate took effect, you have about 2 years before you have to switch over. The whole reason for this ELD mandate is to help law enforcement and drivers as well. Drivers will now be able to have their HOS records on them at all times on their Connect application using the new ELD dashboard, and law enforcement will be able to check these records if need be. The records include everything that is available on their company account on our website. Drivers will also be allowed to login without cell, allowing trips to be saved locally and sent up to the database once cell has gained connection again. Events will be able to be inserted, giving more power to the driver. An example for such a situation is if the driver forgot to logout at the end of his shift. A simple event insertion and a comment will let their supervisor know what happened. This will also set their on-duty, break, time-left timers to the correct values. Every day a driver now must certify their HOS logs, which can also be done in the app, they also have the option to certify any unknown events if there were any and they actually acknowledge them as their own events. There are many great things in the new ELD dashboard on the Connect application, while also keeping the same feel as the HOS dashboard. More information is displayed for clarification, and more functionality is implemented, giving the driver more power, and the supervisors more help.

Specific Expert Essay (Pro400)

InthincHeadless, InthincNote, InthincProvider, InthincVehicle, InthincNoteLib, and SettingsLib are just a few applications that need to be altered for the ELD mandate support of FOB and RFID logins. into effect. The RFID and FOB functionality requires that all of these to communicate with each other correctly. Currently, drivers are not allowed to login with a FOB or RFID if they are HOS enabled, therefore, when swiping a FOB or RFID, we must prompt the driver to login via the Connect app. If they have a tablet connected to the app, we will start the DVIR prompt if their empID matches the one currently logged into the Connect app. The only thing FOB and RFID will be used for right now on the Pro400 are nonHOS logins for LDS customers. Future endeavors might bring another way to use them for HOS enabled drivers, but I doubt that will happen.

When swiping a FOB or RFID, IHFobHandler or NFC will receive the message and hand the number off to InthincHeadless to check various things. It will check whether or not the current RFID or FOB logged in matches the one being swiped in the ProccesFobRfid runnable. If it's the same, it will log the driver out and stop. If it is different, it will log the driver out and log the new one in. Both of these are done by sending a message to InthincVehicle via Json.

The ClearDriverHandler and NewDriverMsgHandler are the two that are used to take care of this process. FobLoginMsgHandler has been removed and now NewDriverMsgHandler is the new hub for FOB/RFID/mobile logins. NewDriverMsgHandler first checks the invalidDriverID and invalidFobRfid list to make sure that drivers that were not setup properly on the portal can't login again. Keep in mind these lists reset every boot cycle, but will still catch them once an Invalid message has been forwarded down to the device when a login is attempted. Depending on what we swipe determines where it will go. If we swipe a FOB or RFID, it will then change the state of the FobRfidLoginRule(which is a state machine ran on a separate thread) by calling doFobRfidLogin(). First state of any swipe will be of the state Fob/Rfid_Lookup. This will just run the LookupEmployee class's run() to search for that employee. The LookupEmployee class is mostly used for HMI devices, but if there is no cell connection and we are nonHOS, it allows drivers to login with as much information that is stored on the device as possible.

After the lookup, the device will send a note through InthincNote using the FobRfid_Info note(231) with the FOB or RFID number and send that message up to the portal. Before any of the information is sent down from the portal via a 1013 or 1014, the device must first successfully broadcast the 231 message. If that message does not send for at least 10 seconds, we cannot verify the login. If the driver is HOS enabled and we cannot verify, we play audio for the user to login with the tablet, otherwise we just log the user in with the information we have. The reason we do this is because nonHOS drivers do not have to worry about ELD laws and any weird info can be sorted out with the company later on if they care. This just allows them to keep track of where the devices are and what ID's were swiped.

Once the commserver actually receives the 231 it will send down a 1013 or 1014 which will be received by the FobRfidInfoHandler. The 1013 will give all of the information associated with that user and a 1014 will indicate the FOB or RFID is not in the database. Once the 1013 is received, it stores all the data and changes the FobRfidLoginRule state to the proper state. It will either be in a cleared state or Fob/Rfid_okay state. If in a clear state, that means there was an invalid driver, otherwise everything was fine and we continue with the login process.

In the FobRfidLoginRule , we check to see if we are HOS enabled, if we are we play audio using TTS, telling the driver to login with the mobile Connect application. If HOS enabled and a tablet is connected, it will start the DVIR prompts that must be answered before logging in, that is if the driver has an empID stored from the 1013. If the driver is nonHOS, it will log them in and also play audio telling them so. These factors are decided by calling isLoginWizardRequired() to check if we are HOS enabled and then depending on that, it will call either doTabletNewDriverFobRfidLogin() or doNewDriverFobRfidLogin(). doNewDriverFobRfidLogin() ends up calling a method in the NewDriverMsgHandler that does nothing but update driverstate information and sends up a NEW_DRIVER message via a 116 note, causing a the device to go into a state ready for user driving. doTabletNewDriverFobRfidLogin() will just check to see if a tablet is connected to the device and start the DVIR prompt, otherwise it will tell the driver to login using a tablet.

All of the above was for logging in with a FOB or RFID, when logging in with a mobile device, it is much simpler at least in this area of the code. NewDriverMsgHandler does nothing but call the ELDTransmitter with the information that was entered on the phone(preTrip,safe,HosJson,empID,etc.) and is taken care of in the new ELD engine system that was implemented for the new mandate. For the FOB/RFID instance of logging in, it will not use the ELD engine. ELD will not use FOB/RFID logins yet until a PO decides its' implementation. For the flow on mobile logins, please refer to the mobile login introduction section.