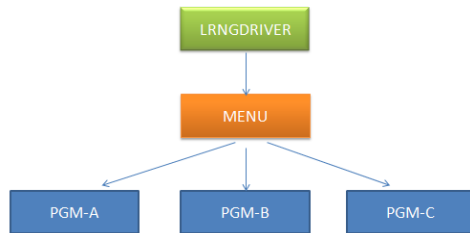


You appear to have structured the App like this:



If you are reusing **existing** 5250 RPG programs you can structure things that way – but there are some guidelines to follow – see

<http://www.longrangemobile.com/docs/LRProgRPG/index.htm#!Documents/anrpgcodereusererepurp.htm>.

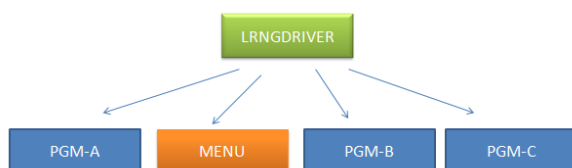
This material might also help you find what your current navigation problem is. Check that when control comes back from sub-program JH\_AGRUMS to the menu JH\_STRBU that it also looks at RequestPROGRAM to see if it should handle the request or just return control back to LRNGDRIVER and let it call the correct program.

---

If you are writing **new** code then as a navigation (menuing) model I would recommend:

(a). Use the one that is included into LongRange (PS: I would be interested in why you need to make up your own menuing structure?).

(b). Create your own - but do not use the classic 5250 “menu program on top” model where the menu calls the sub-menu programs and they always return control to it. Instead treat the menu just like any other program in a ‘flat’ forwards/backwards navigation model – just like web browsers use.



This means that in “MENU” when you decide you need to run “PGM-A” (which is also a formview) you do this:

```
RequestPROGRAM = 'PGM-A';  
RequestACTION  = 'DEFAULT';  
<<Put logical parameters into SavedSTATE>>  
<<return control to LRNGDRIVER>>
```

This means that PGM-A is called by LRNGRIVER rather than by MENU.

You can tell PGM-A what you want it to do by using RequestACTION and putting other logical parameters into the SavedSTATE data structure (typically the current or selected customer, product, order, invoice, etc).

It also means that you can easily go from PGM-A to any other program in LongRange.

If PGM-A decides to redisplay the MENU it does not “return” to the menu - instead it transfers control to MENU – exactly as it would to any other program:

```
RequestPROGRAM = 'MENU';  
RequestACTION  = 'DEFAULT';  
<<Put logical parameters into SavedSTATE>>  
<<return control to LRNGDRIVER>>
```

This model allows the LongRange menu model to jump from any program to any program and it allows you to programmatically transfer control (as opposed to call in a stack) any other program – which is all that a menu really needs to do – maybe the user will come back to the menu – maybe they won't.

You can see this transfer of control in operation in the Employees, Incidents and RPG building Blocks examples. In the Employees demo you display a list (effectively a ‘menu’) of employees. When you touch one the employee details appear – displayed by a different program. If you touch the employees command at the bottom you go back to the “menu” of employees.

When you use this model you might have form views PGM-A, PGM-B and PGM-C (say) that are not directly accessible from a LongRange menu. That's fine – you define them via LR Studio as standalone Form views and set up their commands, tabs, etc. Give them each a symbolic name. When you transfer control to them use the LRNG\_SwitchtoFormView API so that the LR client knows which Form view it is now logically displaying – which allows the commands and tabs to be set up correctly.