Random tick list for the fracture project

No special order – remove when part of the actual plan

//////////////////////////////

7/9/20 – A couple of things here… snow owl data tables, check pagination (some of the pages have bad or double pagination), and using Alan’s newer permission options vs the old cfinvoke calls.

---- older code ---

<cfinvoke component="#application.pathToMainCFCs#.security" method="checkPermission" returnvariable="retCrossCorpPermission\_st">
 <!--- pass in the payee id, corp id, and permission id. --->
 <cfinvokeargument name="payeeId" value="#variables.currentPayeeId#" />
 <cfinvokeargument name="permissionId" value="147" />
 <cfinvokeargument name="corpId" value="#variables.currentCorpId#" />
</cfinvoke>

---- newer code ----

<cfset retCrossCorpPermission\_st = session.permissions.checkPermission(147)>

/////////////////////////////////

7/9/20 – What about folders and themes – for example: top\_secret/secure/ or top\_secret/secure/classic or shop/custom/templates/afb\_clean\_mobile or shop/custom/templates/classic

Do we want to play along the same way or completely change things up? Meaning folders, page flow, processes, validation, login, etc.? Lots of questions here.

What about headers, footers, and black box stuff?

What about secure app stuff vs ecommerce or shop stuff? What structure are we seeking for?

In some ways… I sure would love to redo the whole thing based on our new plan (once it gets finished). However, in the meantime, could we gain by piggybacking on existing logic, flow, folders, and built-in theme options? Great questions.

////////////////////////////////////

7/9/20 – Switching between test and live - All references to cfc’s need to use the shortened generic versions vs the older versions that had specific variables per path – too hard to keep track of

---- sample(s) ----

cfinvoke component="#application.cfcSecurity#” – older strict path style
cfinvoke component="#application. cfcRequests2#” – older strict path style

cfinvoke component="#application.pathToMainCFCs#.security" – new generic way, the variable gets used over and over again
cfinvoke component="#application.pathToMainCFCs#.requests\_2" – new generic way, the variable gets used over and over again

//////////////////////////////////////////

7/9/20 – All cfc’s have a way to call other cfc’s without using Application or session scope vars. Prep for the web/API sockets and using the methods as standalones.

/////////////////////////////////////////

7/9/20 – These are some older projects that we really wanted to get finished before we jumped into fracture… not sure how much weight they carry now…

1. Each file gets a web\_page\_id for storing page or page/user level settings
2. Each file gets cut into blackbox pieces to help with different sections
3. All pages get mini icon menus – we were using the page layouts as the icons. Do we want to keep doing that? Update the icons? Or use something like font-awesome? There are potentially thousands and thousands of pages – which way to go for page navigation?

/////////////////////////////////////////////

7/9/20 – What about code style – script or tags? What about code sign-off, layout, structure, style guide – standardizing things?

What about unit testing, automated testing, other forms of testing?

What about help files, API socket documentation, and other verbiage/help info?

///////////////////////////////////////////////////

7/9/20 – When we start… are we going from a known page (existing) or brand new login? What about the concepts of adilas café or the adilas world stuff? I would love to decouple users from systems and users from corps. Have a general user group and allow those users to go where ever inside the system and even be able to flip/flop servers as needed. Single login type interface.

We have tons of fun prototype stuff for the adilas café idea

///////////////////////////////////////////////////

7/9/20 – Full iceberg style – on everything. Nav, permissions, pages, settings, etc. Hide is the default…

//////////////////////////////////////////////////

7/9/20 – Copying data from data 0 and specific master tables and master table data stuff – similar to the developer’s homepage and the update tables section. What other tables need similar controls?

//////////////////////////////////////////////////

7/9/20 – What about databases and world building options? Allow for shared environments, stacked, single, etc. Universe, galaxy, cluster, solar system, worlds, locations, groups, individuals, data, run all over time. Expand.

/////////////////////////////////////////////////

7/10/20 – On corp-wide settings… Allow main bigger topics to be chosen and then only settings that fall underneath that will be shown. More of the iceberg type model.

Along these same lines… we originally had the idea to use the GPS core to switch modes… data mode, settings mode, permission mode, education mode, etc. It may be nice to revisit those ideas.

Speaking of education mode, there are some super cool prototypes – done by Jonathan Wells – dealing with the education mode.

//////////////////////////////////////////////////

7/10/20 – Talk with Russell, Chuck, Alan, Eric, Wayne, Jonathan, Steve, Dustin, Danny, Bryan, ect. about changes that they want to see (code, design, layout, logic, etc.). Be willing to reach out and get some input from others.

/////////////////////////////////////////////////

7/10/20 – Ice-down date. Currently not really an active piece. We originally had one master date in the corp-wide settings. We were thinking that it might be nice to have ice-down dates for all of the main players. For example: ice-down date for invoices, ice-down for PO’s, etc. Think 12 main players.

////////////////////////////////////////////////

7/10/20 – If we are changing tons of logic… it would be so cool if we could if everything has both transactional and aggregated totals. Right out of the box with clean-up and back-up options for transferring and recording the data. On insert, on update, etc. – triggers, watchers, feeders, etc.

////////////////////////////////////////////////

7/14/20 – The size of the databases, over time, becomes a problem. We may want to look at either splitting up the databases and/or setting up an archive or storage database. It comes down to total record counts and what is actively being queried. Maybe we need to think of the water, snow, and ice methodology. Keep the database quick and snappy on the active and live data… All of the other data needs to be somewhat on ice and/or in a storage type mode.

Small note on the splitting up the databases… this project has also been called separate datasources (dsn’s), world building, bus to motorcycle, get people off the bus.

//////////////////////////////////////////////////

7/14/20 – Cross corp transactions and cross corp functions. Currently we can do some of that because we have multiple corps on the same database. If we split things up, we still need to be able to allow for cross corp transactions and cross corp functions. That either gets tricky or starts playing with temp databases, cross joins, or multiple web/API sockets. We need to keep this in mind as not all corps are just by themselves. We have even seen some corporations that want more and more and more of an enterprise type solution (stackable).

/////////////////////////////////////////////////////

7/14/20 – Graphical homepages with eye candy and charts and graphs. Basic stats and options (jumping off points). Nice and configurable one-pagers. Beautiful and easy to use, configurable dashboards.

We have a bunch of fun stuff from Chuck, Jonathan Wells, and Russell. Lots of fun ideas and concepts.

Perception is believing! Make it look enjoyable but not over the top.

Russell has a sweet dev tool mock-up. It has nice colors schemes, small animations, dynamic one-pagers, buttons and validation, nice forms, loaders, success and error messages, etc. I think Russell calls it his dev tool. He also has a mock-up app that builds things on the fly.

///////////////////////////////////////////////////

7/14/20 – Core pieces with custom views, custom logic, and custom databases – Basically, custom code on bigger and global scale. Russell has a nice plug-in type model that he drew out.

As a fun side note, we did some playing with value add-on cores… 1. Core, 2. Industry, 3. Custom, 4. Business Intelligence, 5. Enterprise Level. See some graphics in the snipping tools folder.

C:\Users\brand\Documents\adilas\_invoices\snipping\_tool\_drawings

/////////////////////////////////////////////////////

7/14/20 – Be able to charge based off of size, storage, media/content, files, images, , database size, years worth of history, usage, API socket calls, etc.

////////////////////////////////////////////////////

7/14/20 – Different views… reports, tabular data, grids, tables, boards, drag-and-drop, charts, graphs, linear data, stacked options, galleries, blogs, feeds, sliders, callouts, accordions, show/hide options, style guides, style sheets, frameworks, etc.

Documentation and visual wireframes. From Russell – Powerful, beautiful, and easy… you could sell it.

/////////////////////////////////////////////////

7/14/20 – We will need a number of different docs… technical, visual, overview, etc.

Talking with Shannon about project management. There is an article out there that talks about getting control and introducing documentation and project management stuff. Here is a link:

Just start in and boil things down. We have tons to work with… just help organize it and boil it down.

https://www.adilascontent.biz/top\_secret/images/corp748\_EA81FBBDE6/pdf/doc\_17364\_C26E92D7E6.docx

//////////////////////////////////////////////////

7/14/20 – build in a way to harvest the users clicks for small tasks and operations. Think of a huge army of clicking ants… what do we need moved or done, add a little light load to every click or certain clicks. Many hands make light work.