Summary report that shows:

Employee, Start date, End date, Regular hours, OT hours, Holiday hours, Sick hours, Total hours.

As a note, the only things that exists are employee, start date, end date, and regular hours as a calculation. All the other pieces we are trying to add in so we can surface them in a report to export.

1. Outline
2. Store the holidays: This would be a good first project and starting place. You would need to claim a new setting. From data 0: “web\_page\_id”. Once you have the id, you can store this in the correct tables. This are samples of some pages that both use a setting and store a setting: “custom/trucking\_home.cfm” “custom/trucking\_home\_settings.cfm”, and “custom/trucking\_settings\_action.cfm”.

The other page you should look at is the duplicate time page: “secure/setup\_reoccurring\_time.cfm” and this has a calendar feature that allows up to a year for setting up holiday days from one year to the next. We will be snagging a piece of this one. User has the power to choose these days/dates.

When storing these settings, we only want to set dates that actually have been selected. If they don’t select it, we don’t want to store it.

“secure/setup\_reoccurring\_time\_action.cfm” is the action page that interprets the calendar with check boxes. Only values that are checked get passed to the next page.

Once id is claimed, you will need to build a page where we can gather that information. We will want to tie it to both manager time clock and admin time clock.

All of the holidays must be setup prior to the date occurring. Won’t retroactively change payroll hours.

Our goal is to use a calendar type interface similar to how we duplicate elements of time and let them set up their holidays via a check box per date. Done on a 24-hour basis. Keep it on a day basis for now. We also want to be able to duplicate from year to year.

We would like to store current year and then next calendar year. Max of 760 days. They would have to adjust the holidays that fall on different days/dates. If we are keeping rolling values, we limit it to two years.

Need to store the information to allow for the duplication and editing process. If they decide tomorrow is a holiday, they can add that in ahead of time.

There are no partial holiday days. Must stick to a per day basis. They choose the day or they don’t.

1. Script the database:

Time Cards- the name of the table is “time\_cards”. It currently has some of these fields: in\_date, in\_time, out\_date, out\_time. There are actually 19 different fields We need to add two more. The first one we’d like to call “time\_card\_flag”. It will need to be a small int (size of 3) with a default of 1. We also need to make sure that it is an unsigned value. We will want to index that field (pretty major search piece).

The second new column will be called: “total\_time” decimal (5,2) with a default of 0.00. We are assuming that all “total\_time” values will be within a 24 hour time period.

We need to make sure to update all existing columns with the time card flag of 1 and the default value of 0.00 for the total time.

Once the new update is done, we will need to create a link on the update homepage so that developers can get to the new database script.

1. Back fill the database

This could be fairly lengthy and may need to be done with some pagination. This is database work with a little bit of logic mixed in. The main goal is to go to all the existing records (technically for every single server) and backfill totals and possible flags.

Here is what the different flags are going to represent.:1=Regular Hours, 2=Overtime Hours, 3=Holiday Hours, 4=Sick Hours, and 5=Needs Manual Attention. Instead of making a new table for this (this list of ids), we are thinking of hard-coding it.

As you write this part of the update, we recommend you build some pagination into the click processes so we aren’t overwhelming the server. Some of these tables may have hundreds of thousands of records. We may want to do a few hundred at a time. \*Brandon may have a helpful sample so please check in with him.

You will need to pull a total record count, figure out how many you want to do per submit (click). Once submitted you will pull just those records in, you will run and do some logic checks, and then update each record based on the logic and criteria. Once you are done with that loop, you will be prompted for the next batch. Once fully finished, you need to show that it is fully finished. This is called an update with pagination.\*\*Brandon said he’s glad to help with this part, as far as what the query looks like, etc.

Just to jog our memory at this time, make sure we cover: what’s the original query going to be, how do we get the counts, how can you tell which part of the process you are in (which page are you on).

1. Make new settings work going forward

a) Some of the logic we use in the backfill will be some of the logic needed in #5. (can’t come in to #5 until you have a good handle on #4)

b) We are going to start at the source where most time cards happen: the user level. The first page is clock in/clock out: “top\_secret/secure/clock\_in\_out.cfm”. The second is: “top\_secret/secure/afb\_timeclock.cfm” We will also have to go to their action pages: “clock\_in\_out\_action.cfm” and “afb\_timeclock\_action.cfm”.

Small checklist of what we are trying to do:
1) If they are starting a new timecard, we automatically set the flag to 1, unless we find that they have set a holiday for that day. If we find it is a holiday, we will automatically set it to a 3. Those are the only auto things we should be doing at the start of a new time card.

2) If they are finishing a time card (clock out), we need to make sure it is within a 24 hour period and if yes, we need to finish the calculation and record the total. If not within 24 hour period, we need to leave the total at 0, and flag with number 5, Needs Manual Attention.

3) Somehow we need to do a little bit of a cleanup. This logic check would only be for regular hours that have been completed. It would be nice to see if these records need to be cascaded into overtime so we would know if we needed regular and overtime pieces.

c) The next level beyond the user level is the managers time clock. To get to the managers time clock: “top\_secret/secure/managers\_time\_clock.cfm”

This allows a manager to start a new time card. You need to have the manager permission and be set up to manage at least one department (Cory can show you this in the demo site).
“top\_secret/secure/clock\_in\_user.cfm” This goes to “clock\_in\_user\_action.cfm”.

Manager can also complete an entire time card (clock in and out with all info filled in). That page is here: “top\_secret/secure/add\_new\_time\_card.cfm”. This goes to “add\_new\_time\_card\_action.cfm”

Manager can add multi clock ins for people in a certain department. That page is here: “top\_secret/secure/multi\_clock\_in\_out.cfm” That page goes here: “top\_secret/secure/multi\_clock\_action.cfm”

Managers will also need to be able to edit time cards in order to fix things. That page is here: “top\_secret/secure/edit\_time\_card.cfm”

All of these will need the new flags to account for regular time, overtime, holiday time, sick time and needs attention time.

d) There are admin functionalities for all of these same things (start, full card, multi, edit). You will need to do everything you just did for manager for admin. They look exactly the same. The only difference is there is one more permission needed. The starting page is: “top\_secret/secure/admin\_time\_clock.cfm”.

Also some cleanup that needs to happen. Older pages with older processes.
There are a bunch of existing reports that pull time cards. It would be nice if it is already calculated to save time. This would clean stuff up. We should be able to get rid of the “just in time calculations” and pull the totals directly from the database. These pages will be lighter and faster and will use the logic and the sums that we already have. Prepping so that we could do payroll for multiple people at once in the future.
Here are some of the reports that will need to be modified:
top\_secret/secure/my\_hours.cfm, top\_secret/secure/managers\_time\_clock.cfm, top\_secret/secure/managers\_time\_clock\_report.cfm, top\_secret/secure/admin\_time\_clock.cfm,

top\_secret/secure/calculate\_payroll.cfm\*\* this one will be tricky.

1. Now we are ready to build the summary report based on new calculations and flagged data.

Name, Date Range 1 and 2, Total Regular, total OT, Total Holiday, total sick, Grand total
and so on for each payee.
2. Export report with snow owl data tables.
3. Testing, code sign off, and deployment.

Questions:

* What happens on a normal process?
* Difference between on the fly calculations and stored data. Trying to go to stored on a per time card basis.
* What will we do naturally and what will base off the new flags and math calculations? Currently we add them all up as if they are regular. Don’t even care about OT until we are ready to do payroll. When we run payroll, we look at a corp-wide setting that tells us how many hours are regular and how many are OT. Nothing is flagged, everything is calculated.
* If we have this natural vs pre flagged scenario, how do we look that up? How are we keeping track of this?
A: When you run the summary report, the logic would calculate overtime (clean things up). Go through and check all timecards. Split off any overtime to its own timecard.
* As they clock in and out, it is currently by department. If someone wants a grand total, do we want to do cross department stuff? Do we keep department details separate or grouped? (maybe not part of this project)
A: Keep it as is for now.
* If we are summing everything up, usually we show a drill down link to show the details.

We currently sort of have this: X hours for normal, and we want to add these: X for OT, X for Holiday. Build in drill downs.
A: We will need to build the drill down info on the admin time clock pages and the manager time clock pages. For now, only an admin report will have the drill downs. The user will not be able to do the drill down.

* How do we know if it needs a special calculation? This may require a whole other page where people can select dates off a calendar.
A: Need a whole other page to set this up.
* It may be easier to have the logic on the front end starting off as regular. Any time someone starts a timecard it starts as regular unless we see specific flags, or