Rough Draft Test Plan

* For the database we believe these functions should be tested
  + Test input and output functionality.
  + Can the database be coded from scratch. – GOOD!!
  + Test the limits on fields in the database. see below.. all part of trying to break it
  + Test for duplicate data being entered in the database. same
  + Test to make sure that participants with the same DOB, address, and other data are tracked so there is no confusion later down the road.
  + Test to make sure privileges for admins are exclusive.
  + Test for security vulnerabilities. please name some.. encryption, certain types of attacks, and what else??
  + Test the database by breaking it.
    - test zipcodes with leading zero (01234)
    - test zipcodes with alpha
    - test zipcodes with fewer than 5 digits
    - test that chris=Chris=CHRIS .. ie. that names can be entered wrong due to capiltalization BUT it does not create duplicate entries
    - test that key fields are not able to be left blank ie. FName, LName, and whatever else we think MUST have an entry… remember, DoB might be blank… or will it? if we force a DoB or make one up if they only have age, then test to ensure two John Smith entries with same made up DoB can’t happen
    - test anytime there is an open text entry point, what happens when you just type tons and tons of lkajfdsljdsafdlfjdsalfjaldskjflkdsajf gobblty gook
    - test what happens when neither M or F are entered for sex
    - test that m=M and f=F for sex… the input team should convert and only store one
    - test phone numbers much the same way you test zipcodes
    - test street numbers similarly .. can you have a 0 for a street number?
* For the intake packet we believe these functions should be tested
  + Test to make sure the intake packet can we exported to excel and printed.
  + Test the fields in the intake packet to make sure they function correctly.
  + Test to make sure all fields are filled. or if they allow blank, that is the desired behavior – AND make recommendations for messages that should be given, like “Are you sure you want to leave the DoB blank.. if you do, a fake DoB will be generated for the purpose of identifying this person as unique from others with the same or similar name”
  + Test to make sure all fields are correctly input.
  + Test to make sure the intake pdf form can be retrieved from the database. This is only if we decide to add the scan of a paper form and include it in the database.
* For the GUI we believe these functions should be tested.
  + Test for functionality of all the buttons on the GUI and make sure they all go to the correct page.
  + Test the functionality of input fields
  + Test the intuitiveness of the GUI, it should be easy to learn. and flow naturally.. should not be anything like iLearn ☺
  + Test to make sure the GUI can connect to the backend database. AND Debug it if it doesn’t work – one thing I didn’t mention to this team is that you will be doing a lot more than just breaking stuff – that’s the fun .. the real challenge, and the reason I think you may actually have the HARDEST job of all the teams, is that you will have to debug stuff and help the developers find their bugs and recommend solutions… this won’t be easy, but I promise it will be rewarding!
  + Test to make sure users can not login into the GUI as an admin.
  + Test to make sure users can ONLY view what they are allowed to view .. ie. John Smith see his record or the other John Smith’s?

I like the idea of having these three categories. I would suggest you create another for Attendance, and one for Reporting. Okay? Thanks.