CIS 322 Test Plan ~ Mintzmyer
Tests inspired by grading rubric: Test setups have prerequisite setups. Test blocks are linear.

Repository Structure
R.Setup: Clone https://github.com/Mintzmyer/CIS322.git, “cd CIS322/import”
and curl https://classes.cs.uoregon.edu//17W/cis322/files/lost_data.tar.gz

R1. Project must contain a lost_archived branch
R1.T1: - Navigate to https://github.com/Mintzmyer/CIS322.git and verify a lost_archived branch
R1.R1: - Under branches, GitHub has ‘lost_archived’

R2. Project must contain a branch for incremental assignment 6, 7, 8, 9, 10
R2.T1: - Navigate to https://github.com/Mintzmyer/CIS322.git and verify assignment branches
R2.R1: - Under branches, GitHub has ‘assignment6’,‘assignment7’, ‘assignment8’, ‘assignment9’, ‘assignment10’.

Installation Functionality
I.Setup: - R.Setup. From CIS322 dir, enter “initdb d”, “pg_ctl -D d -l logfile start”, “createdb lost”, “createdb found”, “createdb test”.

I1. Project preflight.sh must, given an empty DB name, create the needed db tables and install flask code so the app can be run via apache
I1.T1: - From CIS322 dir, enter “bash preflight.sh lost”
I1.R1: - “psql lost”, “\d” Preflight installs DB structure in ‘lost’ db
I1.T2: - From CIS322 dir, enter “bash preflight.sh found”
I1.R2: - “psql found”, “\d” Preflight installs DB structure in ‘found’ db
I1.T3: - From CIS322, enter “./preflight.sh test”
I1.R3: - “psql test”, “\d” Preflight installs DB structure in ‘test’ db

I2. Project must be able to import data from another project implementation
I2.T1: - From CIS322/ “cd import/“, “bash export_data.sh test lost_legacy”
I2.R1: - “psql test”, “\d” DB test now holds all of lost_legacy data

User Interface: Creation/Revokation/Login/Dashboard
U.Setup: - R.Setup. I.Setup.

U1. Project must provide a web service client for creating users. - May not be finished yet
U.Setup: - For now, use create_users page
U1.T1: - Create user with Logistics Officer role
U1.R1: -
U1.T2: - Create user with Facilities Officer role
U1.R2: -

U2. Project must include create user screen that is accessible via “/create_user”
U2.T1: - From web browser, navigate to http://127.0.0.1:8080/create_user
U2.R1: - LOST web app presents create user screen
U2.T2: - Enter a username and password that has not been added by the service client
U2.R2: - LOST web app presents create user screen with error, username does not exist

U2.T3: - Enter a username that has been added by the service client, and create password
U2.R3: - LOST web app automatically logs that user on and presents dashboard

U3. Project must include login screen that is accessible via '/' or '/login'
U3.Setup: Logout of LOST web app, if currently logged in
U3.T1: - From web browser (chrome, firefox, safari, etc), enter http://127.0.0.1:8080/
U3.R1: - LOST web app presents login screen
U3.T2: - From web browser, enter http://127.0.0.1:8080/login
U3.R2: - LOST web app presents login screen
U3.T3: - From *a different* web browser program, enter http://127.0.0.1:8080/
U3.R3: - LOST web app presents login screen

U4. Successful login should route user to a dashboard
U4.Setup: Navigate to http://127.0.0.1:8080/login
U4.T1: - Enter a username and password that has not been added by the service client
U4.R1: - LOST web app presents login screen with error, username does not exist
U4.T2: - Enter an existing username and incorrect password
U4.R2: - LOST web app presents login screen with error, password does not match
U4.T3: - Enter an existing username and correct password
U4.R3: - LOST web app logs that user on and presents dashboard

U5. Dashboard has add facility, add asset, asset report
U5.Setup: Login to LOST dashboard
U5.T1: - Verify user has "Add Facility" button, and click it
U5.R1: - User is taken to http://127.0.0.1:8080/add_facility and app displays Add Facility screen
U5.T2: - Verify user has "Add Asset" button, and click it
U5.R2: - User is taken to http://127.0.0.1:8080/add_asset and app displays Add Asset screen
U5.T3: - Verify user has "Asset Report" button, and click it
U5.R3: -User is taken to http://127.0.0.1:8080/asset_report, app displays Asset Report screen

U6. Dashboard for logistics officer has initiate asset transfers and dispose assets
U6.Setup: Login with user that has "Logistics Officer" role
U6.T1: - Verify user has "Transfer Request" button, and click it
U6.R1: - User is taken to http://127.0.0.1:8080/transfer_req, app displays Transfer Req screen
U6.T2: - Verify user has "Dispose Assets" button, and click it
U6.R2: - User is taken to http://127.0.0.1:8080/dispose_asset, displays Dispose Asset screen

U7. Verify Transfer request, approval, scheduling actions are exclusive to roles
U7.Setup: Create two transfer requests per **O5** test and approve one per **O6** test
U7.T1: - Verify user Dashboard does not prompt Facilities user to schedule approved transfers
U7.R1: - User Dashboard only has transfer request pending approval, not approved one
U7.T2: - Verify user Dashboard does not allow Facilities user to create new transfer requests
U7.R2: - Clicking 'Transfer Request' reroutes to the Dashboard with alert message
U7.T3: - Verify user Dashboard does not prompt Logistics user to approve transfer requests
U7.R3: - User Dashboard has only approved transfer to schedule.

U8. Dashboard for logistics officer has report for transfers needing loading/unloading
U8.T1: - Verify user Dashboard has report for scheduling transfer and click it
U8.R1: - User is taken to http://127.0.0.1:8080/update_transit, displays Update Transit screen

U9. Dashboard for facilities officers has report for transfers needing approval
U9.T1: - Verify user Dashboard has report for approving transfers and click it
U9.R1: - User is taken to http://127.0.0.1:8080/approve_req, displays Approve Request screen

U10. Project must provide a web service client for revoking users. - May not be finished yet
U10.T1: -
U10.R1: -

Organization Functionality: Facilities, Assets, Transfers
O.Setup: - R.Setup. I.Setup. U.Setup.

O1. Project must provide the ability to add new facilities
O1.T1: - Navigate to http://127.0.0.1:8080/add_facility and add Name and Code
O1.R1: - Facility appears in

O2. Project must provide the ability to add new assets
O2.T1: - Navigate to http://127.0.0.1:8080/add_asset, add Tag, Description, Facility and Date
O2.R1: - Asset appears in asset list

O3. Project must provide a report showing which facilties assets are at on a given day
O3.T1: - Navigate to http://127.0.0.1:8080/asset_report, add Date, Facility optional
O3.R1: - Asset appears in report list

O4. Project must provide ability to dispose of assets, only available to Logistics officers
O4.T1: - As logistics officer, navigate to http://127.0.0.1:8080/dispose_asset

O5. Project must allow Logistics Officers to create transfer requests
O5.Setup: Test **U6**
O5.T1: - As Logistics Officer, enter asset tag, current facility, and facility to transfer to
O5.R1: - Request appears in **U8**
O5.T2: - As Logistics Officer, enter different asset tag, current facility, and facility to transfer to
O5.R2: - Request appears in **U8**

O6. Project must allow Facilities Officers to approve transfer requests
O6.T1: - As facilities officer, select transfer request from dashboard and click approve
O6.R1: - Request appears in
O6.T2: - As facilities officer, select transfer request from dashboard and click decline
O6.R2: - Request disappears from all user's dashboards

Testing
T.Setup: - R.Setup. I.Setup. U.Setup. O.Setup.

T1. Project must include a test plan
T1.T1: - Verify this document exists in repo
T1.R1: - This one is a bit tautological.

T2. Project must include test plan results
T2.T1: - Please submit your results!

T2.R1: - This one is on you.