USE CASES

Note: We will define **any user** as the administrative assistant, the department chairs, and the department professors (of the Hendrix College Psychology and Biology departments). **System** refers to the server-side program, and **application** refers to aspects visible from the client side.

Use Case 1: Log in to application
Actor: Any user
Description: Actor successfully logs into application
Preconditions: Actor has navigated to application in browser.
Postcondition: System loads actor's application homepage
Main Success Scenario:
1. Actor inputs username/password into system via login screen text fields, clicks login button
2. System authenticates actor's login information
3. System returns application homepage

Extensions:
2a. Actor login information not authenticated
    2a1. System returns login screen with error message
    2a2. Terminate use case

Use Case 2: View Department funding.
Actor: Any user
Description: View funding distribution for actor's department(s) in real time in Web browser
Preconditions: Actor logged into application
Postcondition: System loads budget data as searchable table.
Main Success Scenario:
1. Actor clicks button to load budget data for their department
2. System loads budget in table form onto client browser

Extensions:
none

__Use Case 3:__ Search Database and Retrieve Information
Actor: Any user
Description: Use keywords to search a database of current budget data/retrieve information
Preconditions: Actor logged into application, viewing budget data
Postcondition: System loads relevant budget data as table
Main Success Scenario:

1. Actor clicks search button
2. System loads search fields
3. Actor inputs search terms
4. System validates input (no SQL injections, etc)
5. System generates database query based on input
6. System queries database
7. System returns query results as table

Extensions:

2a. Actor input is invalid
    2a1. Return relevant error
    2a2. Terminate use case
4a. Query results are empty
    4a1. Reset search fields
    4a2. Alert actor
    4a3. Prompt actor to try again

__Use Case 4:__ Sort Budget Data
Actor: Any user
Description: Sort budget data by specific fields
Preconditions: Actor logged into application, viewing budget data
Postcondition: System loads table of sorted data
Main Success Scenario:

1. Actor Clicks sort button
2. System loads sort fields
3. Actor selects primary sort field
4. Actor selects optional secondary sort field
5. System generates sort query based on selected sort fields
6. System queries database
7. System returns query results as table

Extensions:

5a. Actor selects optional secondary sort field
    5a1. System generates query to first sort by primary field, then secondary
6a. Actor currently viewing part of database
    6a1. System sorts only from most recent query
7a. Query results are empty (should not happen)
    7a1. Reset search fields
    7a2. Alert actor
    7a3. Prompt actor to try again

<u>**Use Case 5:**</u> Hide Fields of Data
Actor: Any user
Description: Hide specific fields of budget data
Preconditions: Actor logged into application, viewing table of budget data, at least 1 column/row not hidden.
Postcondition: Application displays budget table with only relevant fields/entries
Main Success Scenario:

1. Actor clicks hide button on 1+ column(s)/row(s)
2. Application hides data from column(s)/row(s)
3. Application shrinks width of column(s)/height of row(s)
4. Application changes hide button to show button

Extensions:
None

<u>**Use Case 6:**</u> Show Fields of Data
Actor: Any user
Description: Show specific fields of budget data
Preconditions: Actor logged into application, viewing table of budget data, at least 1 column/row hidden.
Postcondition: Application displays budget table with only relevant fields/entries
Main Success Scenario:

1. Actor clicks show button on 1+ column(s)/row(s)
2. Application shows data from column(s)/row(s)
3. Application increases width of column(s)/height of row(s)
4. Application changes show button(s) to hide button(s)

Extensions:
None

<u>**Use Case 7:**</u> View an Archive of Past Budget Data
Actor: Any user
Description: View archived budget data to make predictions and be prepared for cyclic expenses
Preconditions: Actor logged into application
Postcondition: Application displays table of archived budget data from selected cycle
Main Success Scenario:

1. Actor clicks "View Budget Archive" button
2. System loads archive page
3. Actor selects cycle to view from drop down box
4. System loads budget data from selected cycle in table form to browser

Extensions:
4a. No budget data available for selected cycle
    4a1. Notify user
    4a2. Terminate use case

<u>Use Case 8:</u> Add Data Entry to Database
Actor: administrative assistant
Description: Add a data entry into the current budget database
Preconditions: Actor logged into application
Postcondition: New entry added to database and displayed to actor
Main Success Scenario:

1. Actor clicks "Add Data" button
2. System loads adding data page
3. Actor inputs data into fields
4. Actor clicks "Add Entry" button
5. System validates input
6. System adds entry to database
7. System notifies actor of successful entry
8. System adds data entry to main budget table

Extensions:

3a. Actor leaves 1+ fields blank
- 3a1. System notifies actor
- 3a2. System confirms the fields should remain blank
- 3a3. Resume Step 5 of Main Success Scenario

5a. System finds invalid input
- 5a1. System returns relevant error
- 5a2. Terminate use case

<u>Use Case 9:</u> Remove Entry from Database
Actor: administrative assistant
Description: Delete a data entry from budget database
Preconditions: Actor logged into application, viewing table of budget data
Postcondition: Selected entry permanently deleted from database
Main Success Scenario:

1. Actor clicks "remove" button on data entry
2. Application confirms actor's intent to delete entry
3. System deletes entry from database
4. Application deletes entry from webpage budget table

Extensions:

2a. Actor has sudden change of heart, cancels deletion
- 2a1. Terminate use case

__**Use Case 10:**__ Edit Database Entry
Actor: administrative assistant
Description: Edit the fields of a given database entry.
Preconditions: Actor logged into application, viewing table of budget data
Postcondition: 1+ fields of selected entry changed.
Main Success Scenario:
1. Actor clicks "edit" button on data entry
2. Application confirms actor's intent to edit entry
3. System loads edit page
4. Actor edits 1+ fields of entry
5. Actor clicks "make edit" button
6. System validates input
7. Application confirms changes with user
8. System updates database
9. Application returns to page with updated table

Extensions:
2a. Actor has sudden change of heart, cancels edit
  2a1. Terminate use case
6a. System finds invalid input
  6a1. System returns relevant error
  6a2. Terminate use case
7a. Actor has sudden change of heart, cancels edit
  7a1. Terminate use case

__**Use Case 11:**__ View Class-Specific Expenses
Actor: professor
Description: View classes' expenses, including class name/code, purchase, and amount, to plan and prepare
Preconditions: Actor logged into application
Postcondition: Application displays table of expenses for single class
Main Success Scenario:
1. Actor clicks "sort by my classes" dropdown menu
2. Actor selects a class
3. Actor clicks "sort by my classes" button
4. System generates sort query based on selected class
5. System queries database
6. Application loads query result as a table

Extensions:
3a. No class selected
  3a1. Application notifies actor
  3a2. Terminate use case