**Design Document**

**Emerging Tech**

**Staff Schedule**

Jacob Heiser

Contents

1. Introductions  
   1.1. Purpose  
   1.2. Goal
2. Development Environment  
   2.1. Developer Tools  
   2.2. Data Tools  
   2.3. 3rd Party Tools
3. User Experience  
   3.1. Use Cases  
   3.2. User Characteristics  
   3.3. User Interface
4. Production Plans
5. Flow Chart
6. **Introductions**

**1.1. Purpose:**

The schedule optimization tool will be used to optimize the monthly scheduling process. The goal of this tool is to support educational departments efficiently utilize their available faculty.

**1.2. Goal:**

At the current level, many staff members are being over and under-utilized. This tool will help reduce this utilization issue and assign courses to staff appropriately.

1. **Development Environment**

**2.1. Developer Tools:**

>Python

>Spyder

>Jupyter Notebook

>QT Developer

**2.2. Libraries Used:**

> pandas

>sys

> PyQt5

**2.3. 3rd Party Tools:**

>Github

1. **User Experience**

**3.1. Use Cases:**

This application is for use when staff need to be assigned the courses which they need to teach in the upcoming semester. A user will log in to the application, run the program and then input data as requested. Then they will be given the data for the Staff Assignments.

**3.2. User Characteristics:**

The ideal user would be Staff members that are responsible for managing the schedules of other staff members. As such, only Staff with high enough Privileges should be able to access, edit, and send this data.

**3.3. User Interface:**  
 A simple UI that can be understood within a short span of time. Dividing into sperate tabs so each tab represents an objective that trying to be achieved.

**3.3.1 Design Tab:** This is where staff will load each of their files, Academic Report, Daily Numbers, and the Final Schedule that are provided to them. Then they will be able to create the final output in three excel files, “StaffSchedules.xlsx”, “StaffUtilizationReport.xlsx”, and 'OverviewReport.xlsx' excel files with a push of a button.

There is also an Editor Section to edit an already created schedule. Users will need to upload the Academic Report, StaffSchedules, and StaffUtilizationReport files. Then by pushing the editor button will bring up the editor window.

**3.3.2 Schedule Tab:** This is where staff can bring up the final Schedule output StaffSchedules Excel File. By clicking the upload button and selecting the correct file can view it. Users may use as many of the filters as they wish, and using those filters, will find the as many correct entries in the schedule file displayed in the table.

**3.3.3 Utilization Tab:** This is where staff can bring up the final Utilization output, the StaffUtilizationReport Excel File. By clicking the upload button and selecting the correct file can view it. Users may use as many of the filters as they wish, and using those filters, will find the as many correct entries in the Utilization file displayed in the table.

**3.3.4 Overview Tab:** This tab provides a general overview of the current courses that are to be taught. It will allow users to open the OverviewReport Excel File, which will display each Course. Alongside each course will be displayed how many staff members are being fully, over, and under utilized as well as how many people who are currently not teaching a single course but are qualified to teach that course.

**3.3.5 Editor Window:** This window will allow users to edit the now created schedule. There exists one list that contains all the different course codes, one table consisting of all courses being taught, and one table consisting of staff and their utilization status. By double clicking a course, it will filter for all courses with that code and filter all staff that can teach the course.

By selecting a course and a staff member, a user can then use the add button to register the staff to that course. This will override the current staff in charge of the course.

By selecting a course and pressing the remove button, this will now remove any staff that is registered to teach the selected course. Replacing appropriate fields with NA.

By pressing the apply button, this will finalize any changes and regenerate the “StaffSchedules.xlsx”, “StaffUtilizationReport.xlsx”, and 'OverviewReport.xlsx' excel files.

Users can also use the back button to exit and discard any changes they made in error.

1. **Production Plans**

**4.1.1 Coding Logic:**

1. Create Data Frames using data spreadsheets: Academic Appointment Report (AAR), Final Schedule 2023 (FS2023), Copy of 2404 April Daily Numbers 4.8.24 (2404ADN)  
   1. Only import needed information. Excess information can be left out of data frame
   2. (Optional) Create Custom Database - Depending on design could possibly make receiving and using data simpler and easier to update code Would Require additional time and resources
2. Begin by organizing the multiple Courses Available to be taken  
   1. First count all the different Courses available and organize into the appropriate categories
   2. Count the amount of courses in each category
   3. Organize Staff per Category
3. Organize Staff: Locate all Staff that are available to teach that Category
   1. Assign correct Position to Staff: DC, CD, ACD, ADJ
   2. Assign correct Level: Primary, Adjunct + Taught, Adjunct
   3. Then repeat till no more Categories are left
4. Assign courses to the appropriate staff member based off assignment data  
   1. Attempt to place Primary Staff to be assigned in Primary course before they can be considered for other categories: Can be assign elsewhere if no more classes available
   2. Attempt to perform evaluation/equation to determine optimal number of staff needed for Category
   3. If Assigning a staff with title ADJ, they may only teach Online courses not on campus courses
   4. Assign Staff in correct Order on Position first: DC, CD, ACD, ADJ followed by Level: Primary, Adjunct + Taught, Adjunct
   5. If a staff is fully utilized, update their level to level 4 - Reserved. Level 4 will only be used in case there are no more additional staff to support the remaining classes
      1. If staff is labeled as a DC or ADJ, they may only teach a single course. Update DC to level 6 - DC Filled, or level 7 - ADJ - Filled depending on staff title
   6. Program will loop through the staff in the assignment logic a set amount of time, currently 100, before determining that no staff exists. If no staff exists, place in a list for staff errors
   7. If course that is being assign has zero students, do not assign and place in a list for student error
   8. Repeat till no more courses are left to be assigned. If no more staff are available, go to step I.
   9. When level 4 is assigned a class, update to level 5 - Filled. When there are no more level 4 available, reset all level 5 back to level 4. Repeat till no courses are left
   10. Repeat steps for all Categories
   11. See Step 6 for Utilization Logic
5. Build Utilization Report and attempt a fix of any errors  
   1. If maximum amount Staff are fully Utilized or no better optimization is determine to exist (If only underutilize/overutilized exists) then continue to step 7
   2. If a better optimization does exist (If both underutilize and overutilized exists in a single category) then display error to user, show full details, and ask to reassign all categories with this issue
   3. If approved then attempt to rerun the assignment program for the affected categories. If denied then continue to step 7.
   4. After reassignment is completed: show a detailed report, and ask for approval. If denied, repeat step 5c. If approved, continue to step 6.
   5. See Step 6 for Utilization Logic
6. Determine what counts as staff being over, fully, or under-utilzed:

#IF CD/ACDteaching 1 Section >50 = Overloaded

#IF CD/ACD Teaching 1 Section <28 = Underutilized

#IF CD/ACD Teaching 1 Section between 28-50 = Fully Utilized

#Any CD/ACD teaching 2 sections between 25-50 is fully utilized

#Any CD/ACD teaching 2 sections <= 24 is underutilized

#Any CD/ACD teaching 2 sections >= 51 is overloaded

#Any CD/ACD teaching more than 2 sections is overloaded (REMOVED FROM CODE)

#Any CD/ACD teaching 3 sections 24 students or less is fully utilized (CHANGED UNIQUE COURSE TOSECTION)

#AnyCD/ACD teaching 3 sections 25 students or more is overloaded (CHANGED UNIQUE COURSE TOSECTION)  
#Any CD/ACD teaching 4 sections or more is overloaded; everything highlighted in green (CHANGEUNIQUED COURSE TO SECTION)

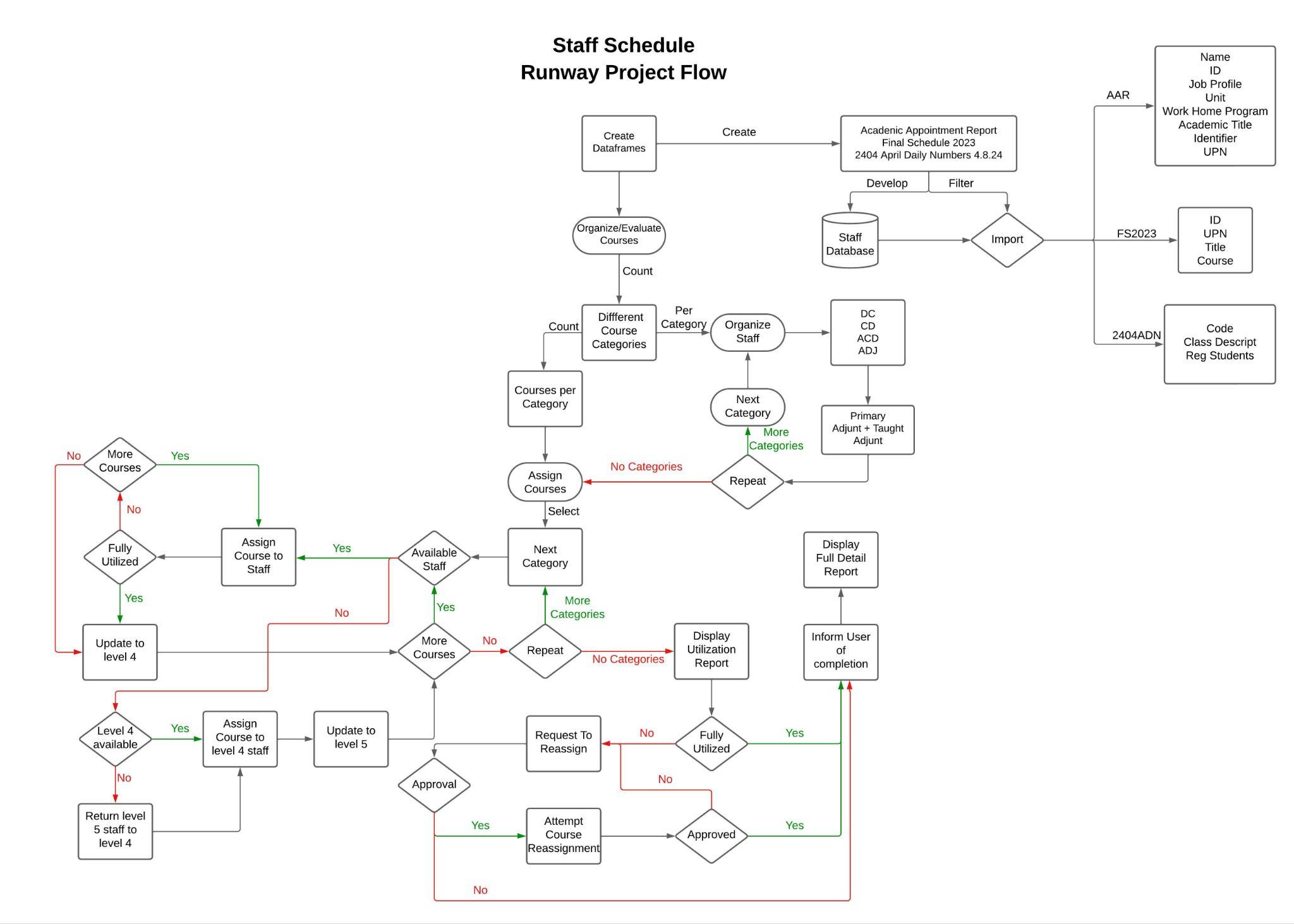
#Any DC/ADJ-Faculty teaching 1 section is fully utilized

#Any DC/ADJ-Faculty teaching >= 2 sections is overloaded

1. Create a general Overview of each of the courses to better visualize which courses are being well optimized
   1. Loop through the created schedule, compare the schedule to the Utilization data.
   2. Count how many Staff are being Fully, Over, and/or Under utilized
   3. Then compare to the Academic Report to see how many staff are not teaching in general
2. Informed the user that Staff has been assigned for their courses to teach. Produce a detailed report, and a downloadable spreadsheets of the data.
3. Allow users to edit the Schedule by uploading the academic report and the schedule and utilization files
   1. Upon loading of editor window, populate the list with the courses, the Schedule table with the uploaded schedule file, and the Staff Table with the uploaded utilization file
   2. If a person clicks a course in the course list, filter tables to matching data.
   3. If a person clicks on a course in the schedule table, save the selected row
   4. If a person clicks on a staff member in the staff table, save the selected row
   5. If adding a staff, overwrite data with new selected staff. If no staff selected – error out
   6. If removing a staff, remove staff entries to appropriate row in the schedule table
   7. When finished, press apply and close window and regenerate output files
   8. If mistake made, push the back button

**4.1.2 Additional Information:**

1. There are three different files that allow the program to work. The Main file: main.py, the Course file: course.py, and Staff file: staff.py. These files spilt the workload to different files for better readability and organization
   1. The main file contains the programming for the UI. It is also responsible for calling functions from the other two files which contain the bulk of the code.
      1. The main also supplies the other files with the correct data that it receives from the user.
   2. The course file has two important functions. One function that organizes data with for the courses and another function that assigns the courses.
      1. counter Function: that organizes data is looking through all course data to get a single instance Code, name, the amount of courses and students. It also looks at the staff to see how many can teach it
      2. assignment Function: that assigns courses, goes through multiple checks to assign staff to each course. Any course that errors out get caught and added to end of the output data
      3. There are some minor functions for formatting data and adjusting current data, consisting of updating the title, updating the code, and a staff’s level status
   3. The Staff file has three functions. One function that get a single instant of each available staff, another function to organize the staff data collected, and the final function to look at how each staff is utilized
      1. counter Function: Is design to look at the staff information and organize the needed information to obtain primary the staff and what they can teach, using the UPN as an identifier.
      2. staff manager Function: Designed to look through the staff and get a single instance of each staff to use later for a more efficient approach
      3. utilization Function: Runs through each schedule looking at each staff member and providing a utilization report.
2. Some of these functions are repeated with different constructors due to data only being obtained in certain portions of the code and obtained in different ways. This way users can update output data without having to restart from scratch each time they start the program and begin editing sooner.
3. **Flow Chart**

****