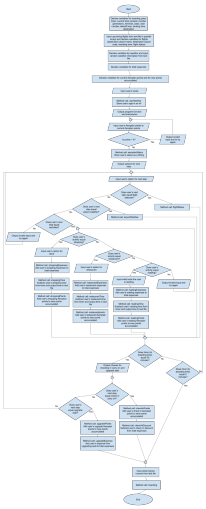
Culminating Programming Project Design

Schedule:

Date	Phase	Task
Dec 8, 2022	Proposal	- Algorithm
Dec 9, 2022		- Programming concepts - Program function
Dec 10, 2022		- Revise and edit proposal
Dec 11, 2022		
Dec 14, 2022	Design	- Schedule - Main method flowchart
Dec 15, 2022		- Method analysis - 3 method flowcharts
Dec 16, 2022		- Revise and edit design
	Programming	
Dec 16, 2022		- Start-up screen - Displaying instructions of the program and how it works - Boarding pass
Dec 19, 2022		- Work on background airport information (flights and weather) - Start on options for activities
Dec 20, 2022		- Aeroplan membership and points - Work on store activity (including expenses) - Start restaurant activity
Dec 21, 2022		- Finish up restaurant activity (including expenses) - Work on magazines (including expenses)
Dec 22, 2022		- Declare and initialize variables for time - Implement times in all the activities - Implement Aeroplan points in all the activities
Dec 23, 2022		- Refine formatting and efficiency of current code - Add comments and documentation to indicate purpose and function of all code
Jan 11, 2023		- Option for user to check in their carry on - Option for user to upgrade their seat
Jan 12, 2023		- Export expenses to text file and calculate percentage of airport's gross income - Boarding time: display activities and expenses of user
Jan 13, 2023		- Work on time variable in activities and troubleshoot any

Date	Phase	Task
		issues with time - Troubleshoot issues with expenses and any other errors
Jan 18, 2023		- Refine internal documentation and efficiency - Test program (by self and others) - Start reflection
Jan 19, 2023		- Complete reflection - Edit and revise reflection and code

Main method flowchart:



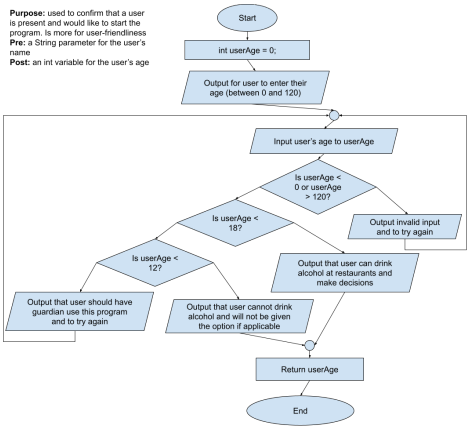
Method analysis:

userStartUp:

Purpose: used to confirm that a user is present and would like to start the program. Function for user-friendliness

Pre (parameters): a String parameter for the user's name

Post (returned value): an int parameter for the user's age



aeroplanStatus:

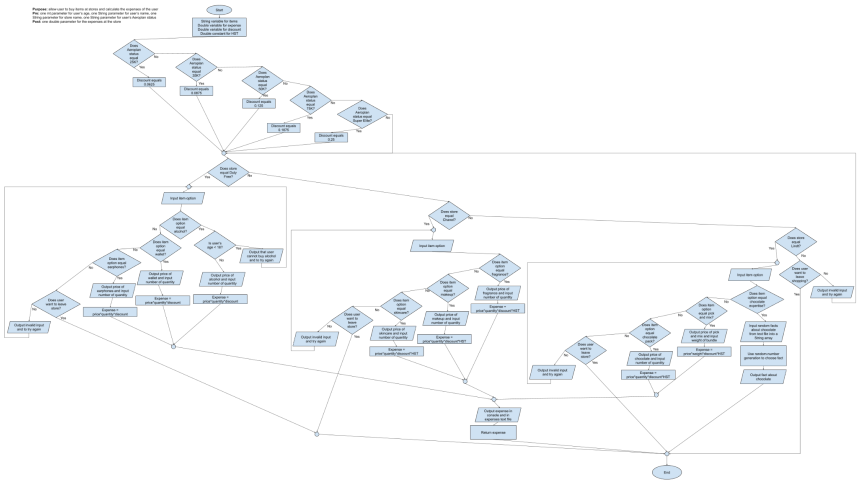
Purpose: determine the user's Aeroplan status from their Status Qualifying Miles, Status Qualifying Segments, and/or Status Qualifying Dollars

Pre (parameters): three int parameters (one for Status Qualifying Miles, one for Status Qualifying Segments, and one for Status Qualifying Dollars)
Post: one String parameter for the user's Aeroplan status

flightStatus:
Purpose: displays incoming flights and their statuses
Pre: one int parameter for the timer, one int parameter for the boarding time hour, one int parameter for the boarding time minute (to determine whether flight has already passed)

airportWeather:
Purpose: displays airport weather
Pre: one int parameter for the timer, one int parameter for the boarding time hour, one int parameter for the boarding time minute (to determine if weather is too late)

shoppingExpenses:
Purpose: allow user to buy items at stores and calculate the expenses of the user
Pre: one int parameter for user's age, one String parameter for user's name, one String parameter for store name, one String parameter for user's Aeroplan status
Post: one double parameter for the expenses at the store



shoppingTime:
Purpose: determine and display the amount of time the user spent at a store
Pre: one String parameter for store name (will obtain the user's expenses from the restaurant expenses text file inside the method to determine the time spent at the store)
Post: one int parameter for the amount of time spent at the store

shoppingPoints:
Purpose: determine and display the number of Aeroplan points that the user obtains from shopping at the store
Pre: one String parameter for the user's Aeroplan status (will obtain the user's expenses from the text file inside the method)
Post: one int parameter for the number of points that the user obtains

restaurantExpenses:
Purpose: allow user to eat at a restaurant and calculate the expenses of the user

Pre: one int parameter for user's age, one String parameter for user's name, one String parameter for restaurant name, one String parameter for user's Aeroplan status
Post: one double parameter for the expenses at the store

restaurantTime:
Purpose: determine and display the amount of time the user spent at a restaurant
Pre: one String parameter for restaurant name (will obtain the user's expenses from the file inside the method and determine time based on number of meals eaten and waitlist)
Post: one int parameter for the amount of time spent at the restaurant

restaurantPoints:
Purpose: determine and display the number of Aeroplan points that the user obtains from eating at a restaurant
Pre: one String parameter for the user's Aeroplan status (will obtain the user's expenses from the text file inside the method)
Post: one int parameter for the number of points that the user obtains

readingExpenses:
Purpose: allow user to read a book and calculate the potential expenses of the user
Pre: one String parameter for book title, one String parameter for user's Aeroplan status
Post: one double parameter for the expenses of buying the book

readingTime:
Purpose: determine and display the amount of time the user spent reading
Pre: one String parameter for the book title (will obtain the number of chapters that the user read from a text file)
Post: one int parameter for the amount of time spent reading

readingPoints: (will only be used if the user buys a book)
Purpose: determine and display the number of Aeroplan points that the user obtains from reading and buying a book
Pre: one String parameter for the user's Aeroplan status (will obtain the user's expenses from the text file inside the method)
Post: one int parameter for the number of points that the user obtains

checkInPoints:
Purpose: determine and display the number of Aeroplan points that the user obtains from checking in their carry on
Pre: one String parameter for the user's Aeroplan status
Post: one int parameter for the number of points that the user obtains

checkInDiscount:
Purpose: determine and display the discount that the user obtains from checking in their carry on
Pre: one String parameter for user's name
Post: one double parameter for the discount that the user has

upgradePoints:

Purpose: determine and display the number of Aeroplan points that the user obtains from upgrading their seat
Pre: one String parameter for the user's Aeroplan status (will obtain the user's discount from the text file inside the method)
Post: one int parameter for the number of points that the user obtains

upgradeExpenses:
Purpose: allow user to upgrade their current seat and calculate expense of user
Pre: one String parameter for user's Aeroplan status, one String parameter for user's seat, one String parameter for user's class, one String parameter for flight number
Post: one double parameter for expense of upgrading the seat

boarding:
Purpose: display user's time and expenses at the airport
Pre: one double parameter for airport's gross income, one double parameter for expenses, one int parameter for points