**Technical Design Document**

Name: Benjamin Durham

Date Created: 1/24/2025

**Program Description**

**# Task at hand**

The exercise asks to create an application to pre-sell a limited number of cinema tickets.

Each buyer will be prompted to enter the amount of tickets they would like to purchase but each buyer can only purchase up to 4 tickets per individual. Furthermore, there are only 20 tickets available for this pre-sale event.

# Programs solution

This program’s purpose is to simulate a real life pre-sale event in which a limited number of tickets can be purchased. The program contains two functions, which our most important function (def ask\_tickets\_and\_validate()) will prompt a user for their input, validate that they have made a valid entry within the confines of the parameters stated above and then display the amount of tickets purchased by how many buyers partook in the pre-sale event.

It was very important to me to push myself and create a solid program that includes validation. I wanted to ensure that not all users will input data just for the sake of the programs success. Meaning, I wanted to create validation that prevented users from inputting a number greater than 4, 0 or less than 0, or even a letter. Areas of improvement, I feel could lie in how the final data is formatted and presented to the user.

**FUNCTIONS USED IN THE PROGRAM**:

1. Function Name: main():
   1. Description: this function runs the program and calls the ask\_tickets\_and\_validate function
   2. Variables: n/a
   3. Logical Steps: calls ask\_tickets\_and\_validate function / runs program
   4. Returns: n/a
2. Function Name: ask\_tickets\_and\_validate():
   1. Description: This function first defines two locals variables. Tickets with an assigned integer value of 20 and lst with an assigned value of an empty list. Then it runs a while loop that will continuously run while the condition of tickets being greater than 0. Inside this while loop is a try/except block for input validation. We first try to get the user’s input and convert that input which will be a default string value to an integer using int(). With that input, the tickets variable will be decremented by input value, the input value will be appended to our empty lst variable, We print the amount of tickets left to purchase to the user. The program then creates a local variable named buyer\_dict that uses dictionary comprehension. The goal is to convert out list into an instance of a dictionary where we have key:value pairs that will represent the buyer and how many tickets they purchased. We use the enumerate() function to iterate or loop over our list and create an index for each item within our list. We use the indices created by enumerate() as our dictionary keys and the associated list elements as the dictionary values. We then create a for loop that will iterate through the indices and values of the buyer\_dict variable and since our list is now a dictionary, we have access to the .items() dictionary method to list out or display the key:value pairs within our dictionary. The for loop will iterate through our dictionary to format and print out the key:value pairs. We use {index + 1} to start our indices at 1 because Python uses 0 based indexing. Our While loop will terminate when the ticket value reaches 0. Underneath this code we have an else statement and except. Our else statement is part of the validation process where it will print an error message to our user if they tried to input a number greater than 4 or is 0. An exception will be thrown to our user if they attempt to enter a non integer value.
   2. Variables: tickets = 20 and lst = [] buyer\_dict, num
   3. Logical Steps:
      * Create local variables
      * Create while loop that runs as long as tickets > 0
      * Get user input and convert to integer from string
      * Store user input in num variable
      * Create logical IF statement that validates user input is less than 5 but greater than 0
      * IF true, decrement num from ticket variable
      * Append num to variable lst
      * Create buyer\_dict variable, assign value using dictionary comprehension
      * Convert lst variable to dictionary and use enumerate() function to create indices and use num as values
      * Create for loop to iterate thru indices and values in buyer\_dict
      * Print and format Buyer# and number of tickets purchased to user
      * IF ticket variable equals 0 break out of while loop
      * IF user input is 0 or greater than 4 print error message to user, while loop continues
      * Throw exception for a ValueError i.e. user attempted to enter a non integer value
   4. Returns: n/a