Step1 :
讀取CSI_data.json讀取dataset裡filepath
將路徑拆分為變數

```
def split_root(file_root):
    parts = file_root.split('/')
    if len(parts) != 6:
        raise ValueError(f"File root is not correctly formatted:
{file_root}")
    return {
        "CLASS_NAME": parts[0],
        "GENDER_COUNT": parts[2],
        "POSITION": parts[3],
        "TIME": parts[4],
        "FILE_NAME": parts[5]
    }
```

Step 2 :
filter 根據題目要求客制 判斷為真時加入output

```
 def filter_data(self):
        filtered = []
        for class_name, files in self.data.items():
            # Show class and entry count
            print(f"Class: {class_name}, Files: {len(files)} entries")
            for file_root in files:
                root_parts = split_root(file_root)
                try:
                    # Pass root_parts, not class_name
                    if self.filter_func(root_parts, file_root):
                        filtered.append(file_root)
                except Exception as e:
                    print(f"Error filtering {file_root}: {e}")
        return filtered
```

sol 1 直接對 *class* 判斷是否包含字串

```
def filter_req_1(root_parts, file_root):
    return "Env3" in root_parts["CLASS_NAME"]
```

sol2 判斷 *F* 出現次數 例如*: F1M1F2* , *F* 出現*2*次因此女生有兩位

```
def filter_req_2(root_parts, file_root):
```

```
    gender_count = root_parts["GENDER_COUNT"]
    # Count the number of females ("F" labels)
    num_females = gender_count.count('F')
    return num_females == 2
```

sol3 直接 出現 *female* 代表只有一位且為女性 直接判斷即可

```
def filter_req_3(root_parts, file_root):
    return "Female" in root_parts["GENDER_COUNT"]
```

sol4 將範圍時間轉換成字串 接著對字串做比較即可

```
def filter_req_4(root_parts, file_root):
    time_value = root_parts["TIME"]
    start_time = "240506_181307"
    end_time = "240507_232434"

    # Check if time_value falls within the specified range
    return start_time <= time_value <= end_time
```

sol5 將上述範例做修改 並加上聯集判斷 即為答案

```
def filter_req_5(root_parts, file_root):
    # Extract the relevant parts
    class_name = root_parts["CLASS_NAME"]
    gender_count = root_parts["GENDER_COUNT"]
    position = root_parts["POSITION"]
    time_value = root_parts["TIME"]

    # Check conditions
    class_name_match = "Env3" in class_name
    gender_count_match = "Male" in gender_count
    position_match = position == "5_posi"
    start_time = "240508_090000"
    end_time = "240508_110000"
    time_match = start_time <= time_value <= end_time

    # Return True if all conditions match
    return class_name_match and gender_count_match and position_match
and time_match
```