Part B

Q1.

Assume both of dataset 1 and dataset 2 has been uploaded to dataframes (df1 and df2) in pandas.

For both of dataset 1 and dataset 2 (df1 and df2), as I plan to load the datasets into a dataframe in pandas, and pandas doesn't work well with the column names which contains spaces, so I will replace the spaces with an underline ('_').

```
df1.columns = [c.replace(' ', '_') for c in df1.columns]
df2.columns = [c.replace(' ', '_') for c in df2.columns]
```

For, dataset 1 (df1), as we want to find a relation with the operators and the devices, so the data which misses *Device ID* or *Operator* is useless, so I drop the rows with a NaN in Operator or Device ID.

```
need_to_dropna_col = ['Device_ID', 'Operator']
df1 = df1.dropna(subset = need_to_dropna_col, axis = 0)
```

According to the required task, we only need *Device ID*, *Quality Tested Date/Time* and *Operator*, so I redefine the df1 and remove the duplicate rows.

```
df1 = df1[['Device_ID', 'Quality_Tested_Date/Time', 'Operator']]
df1.drop_duplicates()
```

For, dataset 2 (df2), as we need to identify the devices by Device ID, so Device ID must be uniform and exists. Therefore, I drop the data without Device ID.

```
df2 = df2.dropna(subset = ['Device_ID'], axis = 0)
```

According to the required task, we only need *Device ID* and *Support Ticket Date/time*, so I redefine the df2 and remove the duplicate rows.

```
df2 = df2[['Device_ID', 'Support_Ticket_Date/Time']]
df2.drop_duplicates()
```

Finally, as we need to conduct the required task, so I need to join dataset 1 and dataset 2 on their *Device ID*. Because I must keep dataset 1 integrity, so I choose to use LEFT JOIN.

```
newdf = df1.merge(df2, on='Device_ID', how='left')
```

Q2.

A)

I'm going to use K-means to solve the problem. Because the owner has no fixed target for the number of groups, which is an unsupervised problem that needs to find some relations in an entire dataset, therefore K-means will be the best choice. It will help the owner to group similar data into clusters (group).

B)

1. Choose a value of k. Assume we choose 3 for k, then each data point will be assigned to one of three clusters randomly.

2. Computer the centroids for each of the clusters, then reassign each data point to the nearest centroids.

3. Repeat step 2 to compute the new centroids until there is no switching of points from one cluster to another. In this way, all customers are grouped.

C)

I will build a K-means model in python by sklearn, and change the value of n_clusters. In the meanwhile, we need to calculate the sum form each data point to its assigned centroids. Then, the value of k with the minimum sum is the number of groups.

Q3.

N1 is 10, N2 is 90 and N3 is 10.
Because the cross-validation is 10-Fold, so we have ten folds. Thus, N1 is 10. As in the k-fold cross-validation, for every i = [1, k], the model is trained on every fold except the ith fold and computes the test error on the ith fold. Because there are ten (#examples/k = 10) examples for each fold, so N2 equals 90 (#examples_in_each_fold * (k - 1) = 90) and N3 equals 10 (#examples – N2 = 10).

Q4.

As precision = TP / (TP + FP), precision = 8 / (8 + 2) = 0.8
As recall = TP / (TP + FN), recall = 8 / (8 + 12) = 0.4
As F1-score = 2 * (precision * recall / (precision + recall)),
so F1-score = 2 * (0.8 * 0.4 / (0.8 + 0.4)) = 0.53

Q5.

A)

I will advise them to use OAuth, it can identify user accounts and grant proper permissions. For example, a user wants to log into a system, then the system will request authentication through a token. After granting the request, the user will send the token back and the system will send a token with expiry time to an authentication server. As the token can be revoked after a certain time, OAuth is much secure.

B)

I will advise them to use API Key, because it can get usage analytics and limit the usage rate. For example, API Key can revoke the API key if the client violates the usage agreement, which makes the system secure. And the API Key can return HTTP response code 429, if requests are coming in too quickly, which limits the usage rate.

Q6.

The response will contains a status line, a header and a body.

```
Status line  --  HTTP/1.1 200 OK
Header       --  Date: XXXX Content-Type: XXXX ….
Body         --  {
                     “resultSize”: XX,
                     “results”: [{
                                  “id”: XX,
                                  “type”: “latte”,
                               },{XXXX},…
                             ]
                 }
```

Q7.

A)

I will use random forest to solve this problem. As we need to predict who is at risk and we already have some data that the machine can learn from it, so it’s a classification problem. Because our training data has a higher dimension, so random forest will produce a good result. However, because random forest is quite complex, so it will be slow.

B)

I will use *Identifier of people close to device*, *Distance* and *Duration of contact* in table 1 and *Identifier of people* and *Infection tested* in table, and scale *Distance* and *Duration* in table 1. Because *GPS Location* is replaced by *Distance* and *Timestamp* is replaced by *Duration of contact*, so we don’t need it.