All below is copied from the assement page head to page 10 on this document or where it says Requitements

# IT5016D: Software Development Fundamentals

### Assessment 2: Software Project

*This assessment represents 50% of your final grade.*

### **Assessment Overview**

Think about the Software Project like a piece of art, only you're creating a *Software Project* based on your own technical research and emerging practice.

### **Learning Outcomes**

**LO1:** Develop program logic to a professional working brief.

**LO2:** Apply the stages of software development to develop a simple code solution.

### **Conditions**

* It is recommended that you spend between **20 and 25 hours** on this assignment. It is better to  
  add content as you find it, rather than at the end.
* All course materials, and any other resources, can be used to complete this assessment.
* The work you submit must be your own work. It is an individual assessment.
* You can ask your facilitator to clarify the instructions, and/or for advice, but they cannot do or solve the assessment tasks – you must carry out the tasks yourself!

### **Assessment Instructions**

**This assessment gives hands-on experience of the language of the Software Development Fundamentals.**

### 1. Please read the scenario below and apply code solutions to develop a Help Desk ticketing system prototype.

2. Record the stages of the Software Development Life Cycle (SDLC) and include with the submission of your Software Project.

*It is valuable to record your experiences during the Software Development Lifecycle (SLDC), which is why we ask you to submit this with your Software Project. These experiences will help you to professionally communicate and contextualize your practice as a professional and build confidence in your technical skills and practice.*

#### **Scenario**

**Help Desk Ticketing System Prototype**

**Client Requirements**

The client would like a Help Desk ticketing system prototype developed.

The Help Desk ticketing system should handle tickets from **internal customers only**.

Tickets will be requested for assistance **from** the Help Desk **by** staff members of the organisation.

**Requirements of the Help Desk Ticketing System**

**Tickets:**

* Tickets can be submitted by providing all of the following information:
* Staff ID,
* Ticket creator name,
* Contact email
* Description of the issue

Internal Tickets’ ticket number should be assigned automatically using the **counter static field plus 2000.**

All information must be provided as input while submitting the ticket.

**Responding to tickets:**

* If the ticket’s description of the issue contains the words ***“Password Change”,*** the new password should be generated following the rule,
* T*he first two characters of the staffID, followed by the first three characters of the ticket creator name.*

**Hint:** (can be useful to consider: split(), join(), string operations)

* There should be an option, after the ticket has been submitted, to **respond** to a ticket by providing a feedback response.

**Default response** can be set as “Not Yet Provided”.

**Statistics:**

* There should be a way to keep track of:
* The number of tickets submitted
* The number of resolved tickets
* The Number of open tickets
* A way to display those statistics to the console.

* If the **staff member has submitted** the “Password change” request, after the new password is generated and the ticket’s response has been updated, the ticket should close, with the number of closed tickets increased and the number of open tickets reduced by 1. Ticket’s status should be changed to “Closed”.

* Once a member of the **IT department provides the response to a ticket**, the ticket should close, with the number of closed tickets increased and the number of open tickets reduced by 1. Ticket’s status should be changed to “Closed”.

* There should be an option for the **IT department to reopen the ticket**. At this point the number of open tickets should be increased and the number of closed tickets should be reduced by 1. Ticket’s status should be changed to “Reopened”

**Displaying the ticket:**

* There should be a way to display the ticket information:
* Ticket number,
* Name of the ticket’s creator,
* StaffID,
* Email address,
* Description of the issue,
* Response from the IT department and ticket status (open, closed or reopened).

*The output format is shown in the examples at the end of Technical Requirements section.*

**Technical Requirements**

The senior developer has provided you with the following technical requirements for the project.

* The Ticket class should contain common ticket information in the Ticket class.
* The Ticket class should also have method allowing the staff to submit ticket and the IT team to respond to the tickets, specifically resolve, reopen and provide a response to the ticket.
* The Ticket class should contain a method for resolving password change requests. As well as calling the method that would generate the new password, it should set up a response for the ticket and change the ticket status to closed.
* There should be a method to print information for all the ticket objects.

**Hint:** research and use List<Ticket>

* The TicketStats method in Ticket class should contain information on ticket statistics and shall be able to return the statistics information.
* The main class, containing the Main method.
* Create at least one instance of submitting tickets and include at least one ticket with the request of “Password change”.
* After the tickets are created, display ticket statistics.
* Resolve some of the tickets, then display the ticket information and ticket statistics. o Reopen some of the resolved tickets, then display the ticket information and ticket statistics.

* **The example output is provided below:**

**Displaying Ticket Statistics**

Tickets Created: 3

Tickets Resolved: 1

Tickets To Solve: 2

**Printing Tickets:**

Ticket Number: 2001

Ticket Creator: Inna

Staff ID: INNAM

Email Address: [inna@whitecliffe.co.nz](mailto:inna@whitecliffe.co.nz)

Description: My monitor stopped working

Response: Not Yet Provided

Ticket Status: Open

Ticket Number: 2002

Ticket Creator: Maria

Staff ID: MARIAH

Email Address: [maria@whitecliffe.co.nz](mailto:maria@whitecliffe.co.nz)

Description: Request for a videocamera to conduct webinars

Response: Not Yet Provided

Ticket Status: Open

Ticket Number: 2003

Ticket Creator: John

Staff ID: JOHNS

Email Address: [john@whitecliffe.co.nz](mailto:john@whitecliffe.co.nz)

Description: Password change

Response: New password generated: JOJoh

Ticket Status: Closed

**Displaying Ticket Statistics**

Tickets Created: 3

Tickets Resolved: 2

Tickets To Solve: 1

**Printing Tickets:**

Ticket Number: 2001

Ticket Creator: Inna

Staff ID: INNAM

Email Address: [inna@whitecliffe.co.nz](mailto:inna@whitecliffe.co.nz)

Description: My monitor stopped working

Response: The monitor has been replaced.

Ticket Status: Closed

Ticket Number: 2002

Ticket Creator: Maria

Staff ID: MARIAH

Email Address: [maria@whitecliffe.co.nz](mailto:maria@whitecliffe.co.nz)

Description: Request for a videocamera to conduct webinars

Response: Not Yet Provided

Ticket Status: Open

### **Submission Checklist**

|  |  |
| --- | --- |
| **Task** | **Evidence Required** |
| 1 | *Software\_Project* Python Code file |
| 2 | *ReadMe* file with instructions |
| 3 | Word/PDF file with the details of Software Development Lifecycle stages during the development of the project |
| 4 | Instructions required to display your software project |

### **Success Criteria**

The assessment will be marked according to the rubric below. Please take time to read it and ask for clarification if necessary.

You are required to obtain a minimum of 50% of the total available marks to be successful in this assessment.

A maximum of two attempts are allowed to complete this assessment. The maximum percentage to be awarded on a second assessment attempt is 50%.

### **Submission instructions**

When you have completed the assessment:

* Verify that all items have been completed using the submission checklist provided
* Navigate to the assignment item in Canvas (Dashboard > Select *Course > Assignment* tab > ***Assessment 2: Software Project***).
* Please ensure your document is named "*IT5016D\_Assessment 2\_<StudentID>*".
* Upload your assessment document using upload facility in Canvas
* Read the declaration and submit the assessment

### **Declaration**

By submitting your work, you are indicating that you agree to the following declaration:

**"The work presented in this assessment is to the best of my knowledge original, except as acknowledged in the text, and the material has not already been submitted, either in whole or in part, for any academic award at this or any other tertiary institution. I promise not to share this assessment in part or whole with any other student at Whitecliffe or outside this campus."**

IT5016: Assessment 2 - Software Project

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| This criterion is linked to a Learning Outcome Software Development Life Cycle (SDLC) | |  |  |  |  | | --- | --- | --- | --- | | 20 to >16.0 pts  Appropriate stages of the software development life cycle were followed and includes a description of each stage (requirements analysis and solution design, solution development, testing). | 16 to >13.0 pts  Some stages of the software development life cycle were followed and includes a description of some of these stages. | 13 to >10.0 pts  General indication the appropriate stages of the software development life cycle were followed with little description of any of the stages. | 10 to >0 pts  Little or no attempt to follow the appropriate stages of the software development life cycle and/or describe any of the stages. | | 20 pts |
| This criterion is linked to a Learning Outcome Solution Development: Code to re-set password | |  |  |  |  | | --- | --- | --- | --- | | 20 to >16.0 pts  Code to reset the password correctly reset the password using String Operations. | 16 to >13.0 pts  Code to reset the password shows correct use of String Operations to some extent and produces a password following the client requirements specifications. | 13 to >10.0 pts  Code to reset the password shows correct use of String Operations but does not produce a password following the client requirements specifications. | 10 to >0 pts  The code to reset the password does not show the correct use of String Operations and does not produce a password following the client requirements specifications. | | 20 pts |
| This criterion is linked to a Learning Outcome Solution Development: Statistics | |  |  |  |  | | --- | --- | --- | --- | | 20 to >16.0 pts  Object Oriented concepts are used correctly in the solution development. | 16 to >13.0 pts  Object oriented concepts are used in the solution but are not implemented in the most efficient way. | 13 to >10.0 pts  Minimum Object-oriented concepts are used in the solution development. | 10 to >0 pts  Object-oriented concepts are not used appropriately. | | 20 pts |
| This criterion is linked to a Learning Outcome Object-oriented concepts are not used appropriately. | |  |  |  |  | | --- | --- | --- | --- | | 20 to >16.0 pts  Evidence of technical requirement analysis and solution design stages match the specifications of the brief and more than one ‘Classes’ are used in the solution development. | 16 to >13.0 pts  Evidence of technical requirement analysis and solution design stages match the specifications of the brief. | 13 to >10.0 pts  Evidence of technical requirement analysis and solution design stages partially match the specifications of the brief. | 10 to >0 pts  Evidence of technical requirement analysis and solution design stages is not provided and/or the evidence does not match the specifications of the brief. | | 20 pts |
| This criterion is linked to a Learning Outcome Solution Development: Compile and run | |  |  |  | | --- | --- | --- | | 20 to >15.0 pts  The solution created compiles and runs as expected. | 15 to >10.0 pts  The solution created compiles but does not run as expected. | 10 to >0 pts  The solution created does not compile. | | 20 pts |
| Total Points: 100 | | | |

For this project I'm going to be using the waterfall model

What you see above is the info that was given for the project

# Requitements

**Tickets**

Tickets need to have the following information when submitted by user

1. Staff ID
2. Ticket creator name
3. Contact email
4. Description of the issue

When I ticket is made it needs ticket number (UID) assigned automatically starting at 2000

## Responding to tickets

If the ticket’s description of the issue contains the words ***“Password Change”,*** the new password should be generated following the rule. T*he first two characters of the staffID, followed by the first three characters of the ticket creator name.*

There should be an option, after the ticket has been submitted, to **respond** to a ticket by providing a feedback response (**Default response** can be set as “Not Yet Provided”)

## Statistics

There should be a way to keep track of:

1. The number of tickets submitted
2. The number of resolved tickets
3. The Number of open tickets
4. A way to display those statistics to the console

If Description of the issue is ***“Password Change”***, then generate new password for the response. Also, the number of open tickets decreased by 1 and closed increased by 1. change ticket status to “closed”

If a member of the IT department provides the response to a ticket, then the number of open tickets decreased by 1 and closed increased by 1. change ticket status to “closed”

If a member of the IT department reopens a ticket, then the number of open tickets increased by 1 and closed decreased by 1. change ticket status to “reopened”

## Displaying the ticket information

There should be a way to display the ticket information e.g:

1. Ticket number
2. Name of the ticket’s creator,
3. Staff ID
4. Contact email
5. Description of the issue
6. Response from the IT department
7. ticket status (open, closed or reopened)

Example of what It might look like:

Tickets Created: 3

Tickets Resolved: 1

Tickets To Solve: 2

**Printing Tickets:**

Ticket Number: 2001

Ticket Creator: Inna

Staff ID: INNAM

Email Address: [inna@whitecliffe.co.nz](mailto:inna@whitecliffe.co.nz)

Description: My monitor stopped working

Response: Not Yet Provided

Ticket Status: Open

## Technical Requirements

The senior developer has provided the following technical requirements for the project.

* The Ticket class should contain common ticket information in the Ticket class.
* The Ticket class should also have method allowing the staff to submit ticket and the IT team to respond to the tickets, specifically resolve, reopen and provide a response to the ticket.
* The Ticket class should contain a method for resolving password change requests. As well as calling the method that would generate the new password, it should set up a response for the ticket and change the ticket status to closed.
* There should be a method to print information for all the ticket objects.
* The TicketStats method in Ticket class should contain information on ticket statistics and shall be able to return the statistics information.
* The main class, containing the Main method.
* Create at least one instance of submitting tickets and include at least one ticket with the request of “Password change”.
* After the tickets are created, display ticket statistics.
* Resolve some of the tickets, then display the ticket information and ticket statistics. o Reopen some of the resolved tickets, then display the ticket information and ticket statistics.

Design:

## Logical design:

Ticket class

common information

1. staff\_ID
2. creator\_name
3. Contact email
4. Description of the issue
5. UID for each ticket made
6. Response (**Default response** can be set as “Not Yet Provided”)
7. Ticket Status (open, closed or reopened) default open
8. ID/count for the UID of the next ticket (this can be done with a static value in the class the all classes have)
9. Ticket\_list List for storing the tickets
10. Maybe some values called open and closed to keep track of amount open and closed

Methods

1. submit ticket (this will be done with \_\_init\_\_)
2. Change\_password: If Description of the issue is ***“Password Change”, do*** change Ticket\_Status to “closed” and response to the first 2 chars of staff\_ID and Frist 3 chars of creator\_name
3. Open: Ticket\_Status to “open”
4. Reopen: Ticket\_Status to “reopened”
5. Close: Ticket\_Status to “closed”
6. Resolve :ticket\_status to “closed” and change response
7. ticket info all: retuns every ticket in Ticket\_list
8. ticket info: retuns Ticket information as a dic
9. TicketStats: retuns total, open, Closed number of tickets

Other

After the tickets are created, display ticket statistics.

Create at least one instance of submitting tickets and include at least one ticket with the request of “Password change”.

After the tickets are created, display ticket statistics

Resolve some of the tickets, then display the ticket information and ticket statistics. Reopen some of the resolved tickets, then display the ticket information and ticket statistics.

The main class

Methods

1.main: where the main loop of this program is and where everything starts and ends

System for picking what to do e.g.

1. Make a ticket
2. Print a stats of tickets
3. Print ticket information (id given)
4. Print all ticket information and stats
5. Add response to ticket
6. Close a ticket (id given)
7. Reopen a ticket (id given)
8. Exit

## Physical design:

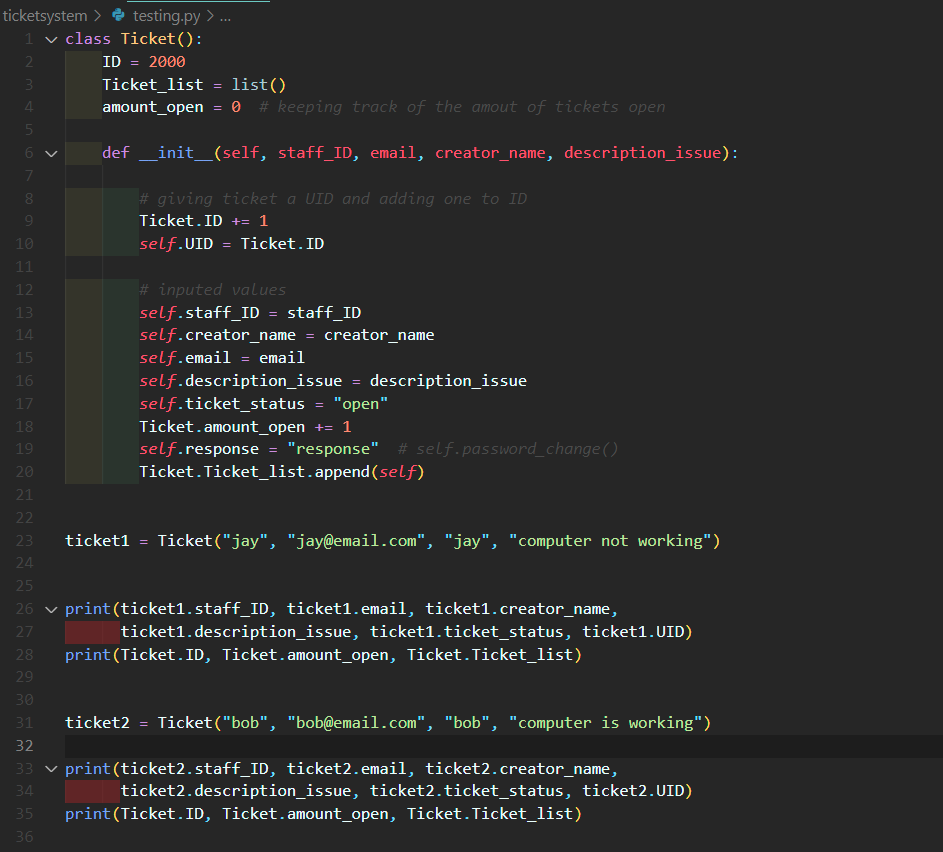
Any device that can run python 3.10 or above will run this program

Testing:

Testing making a ticket using the ticket class:

Making 2 tickets then printing the results

For this I have also commented out password change function witch will be tested later on.



Expected result:

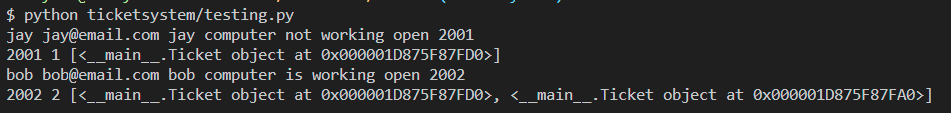
jay jay@email.com jay computer not working open 2001

2001 1 (one class object in a list)

bob bob@email.com bob computer is working open 2002

2002 2 (two class object in a list)

Here is the output

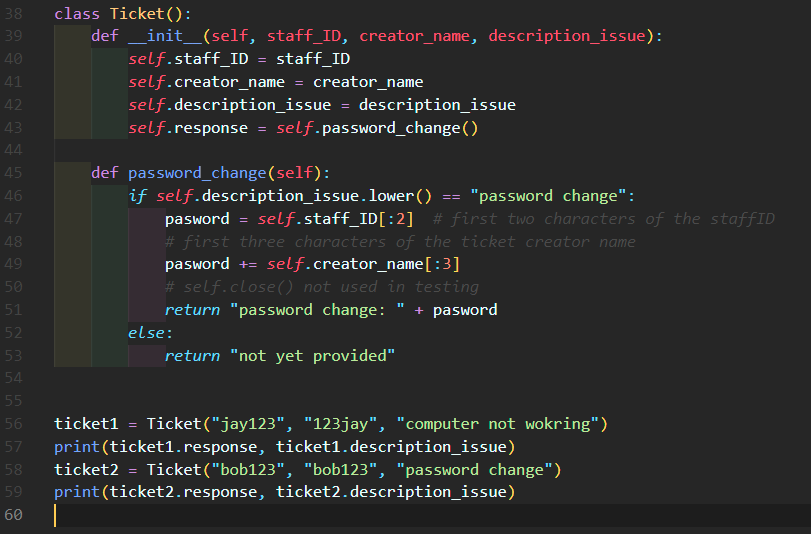


Good this means that making a ticket using this class is ok nothing to fix

Testing pasword change

I have the minim for testing

For this I have also commented out close function witch will be tested later on

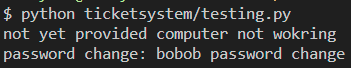


Expected result:

not yet provided computer not wokring

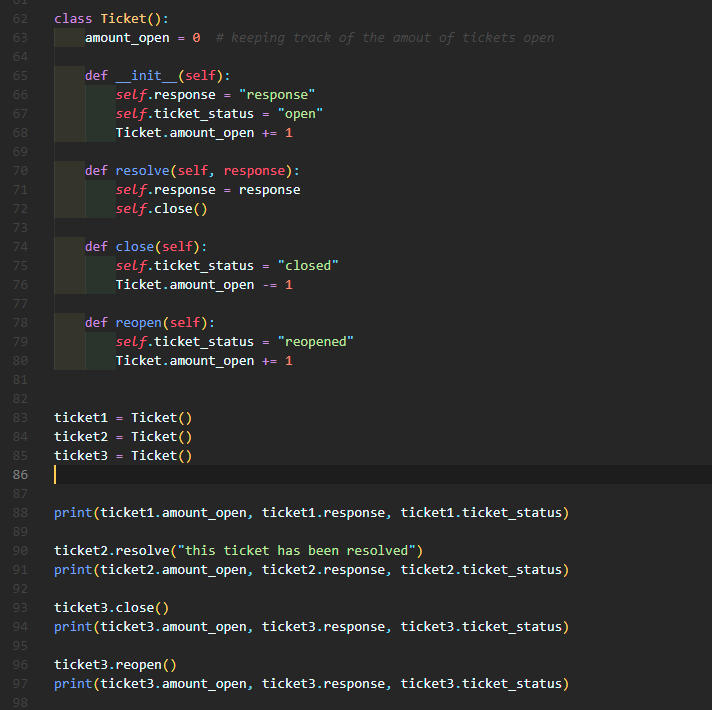
password change: bobob password change

Result:



Expected and result are same. nothing to change here

Testing open, resolve, close, and reopen fuctions



Expected result:

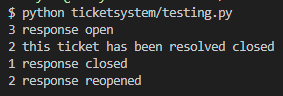
3 response open

2 this ticket has been resolved closed

1 response closed

2 response reopened

Result:



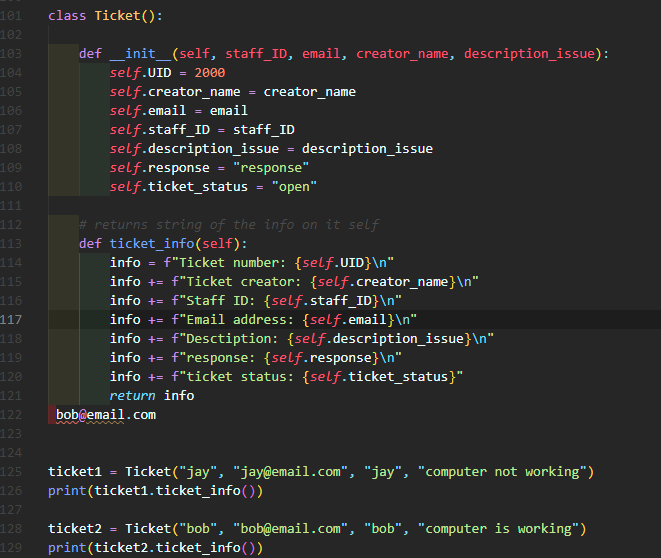
Expected and result are same. nothing to change here

Checking info



When reading over the code I forgot to put email in the info

New code



Ticket number: 2000

Ticket creator: jay

Staff ID: jay

Email address: jay@email.com

Desctiption: computer not working

response: response

ticket status: open

Ticket number: 2000

Ticket creator: bob

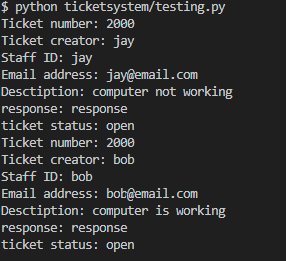
Staff ID: bob

Email address: bob@email.com

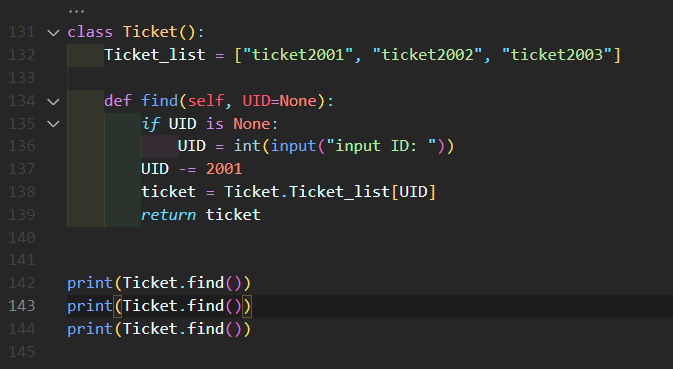
Desctiption: computer is working

response: response

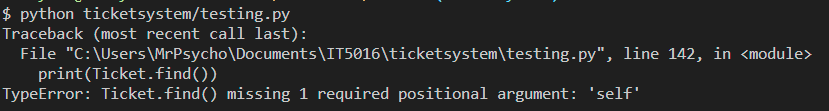
ticket status: open



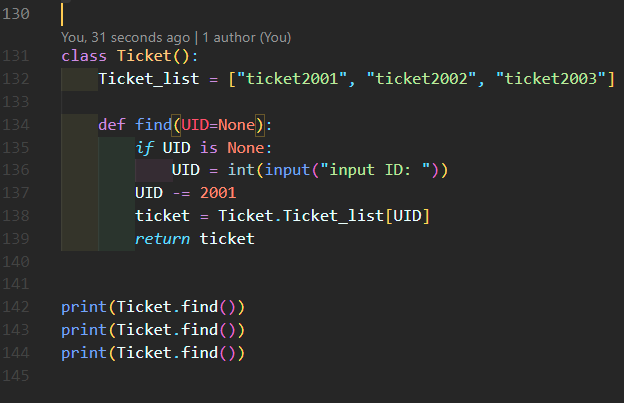
Expected and result are same. nothing more to change here



Error don’t need self for testing need in code



New code



expected

input ID: 2001

ticket2001

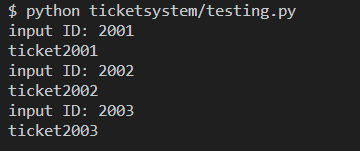
input ID: 2002

ticket2002

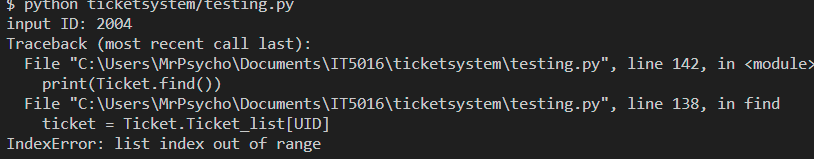
input ID: 2003

Ticket2003

Result:

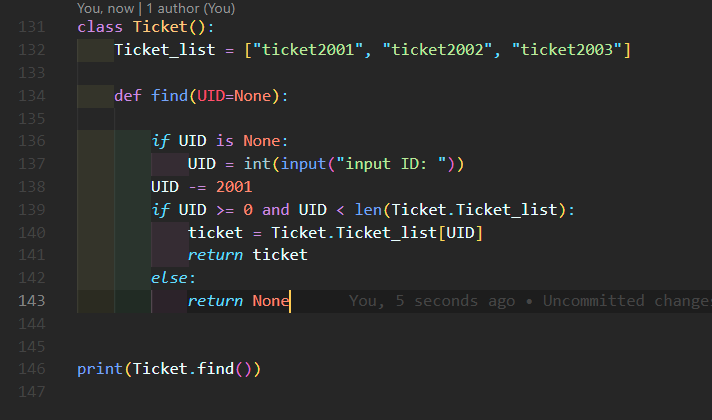


Now what about if there is no ticket for that id



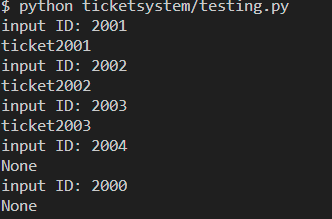
Index error so we have to add some code to check if the ticket is there

New code



We check if UID is greater than and equal to 0 and UID is less than the length of the ticket list then get the ticket and return it

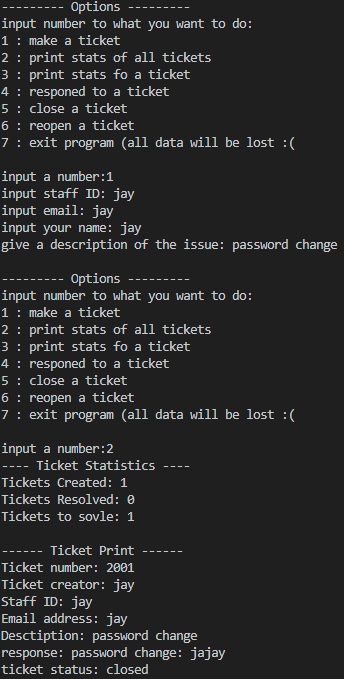
Else we return None “null” if there is no ticket with that UID



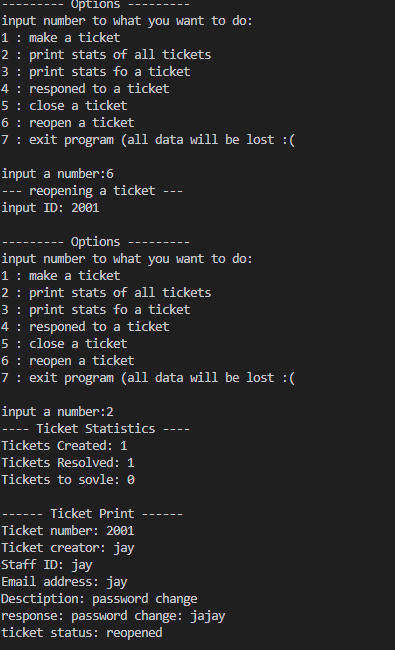
Fixed and done

Full testing

Make a ticket with password change



Ticket reopened



Ticket closed

