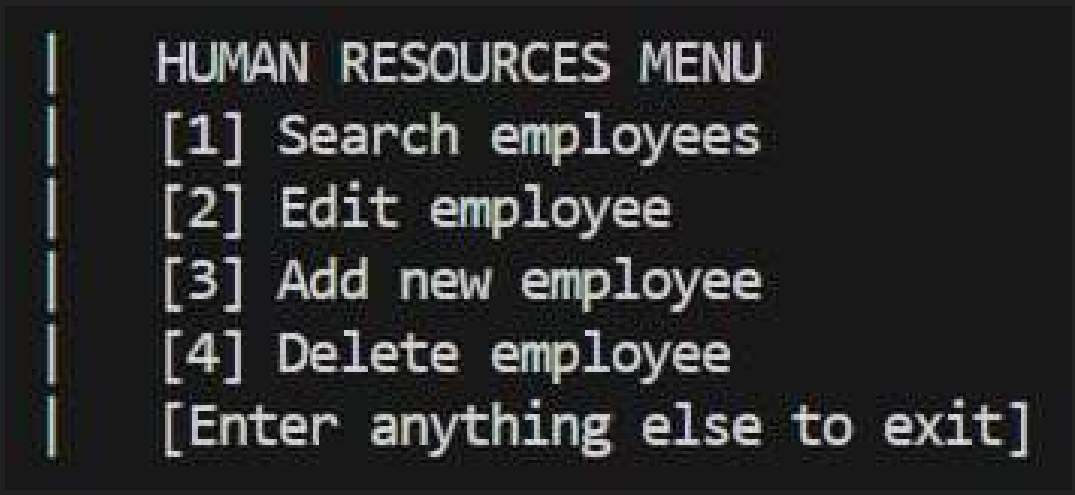
T1A3: Terminal Application

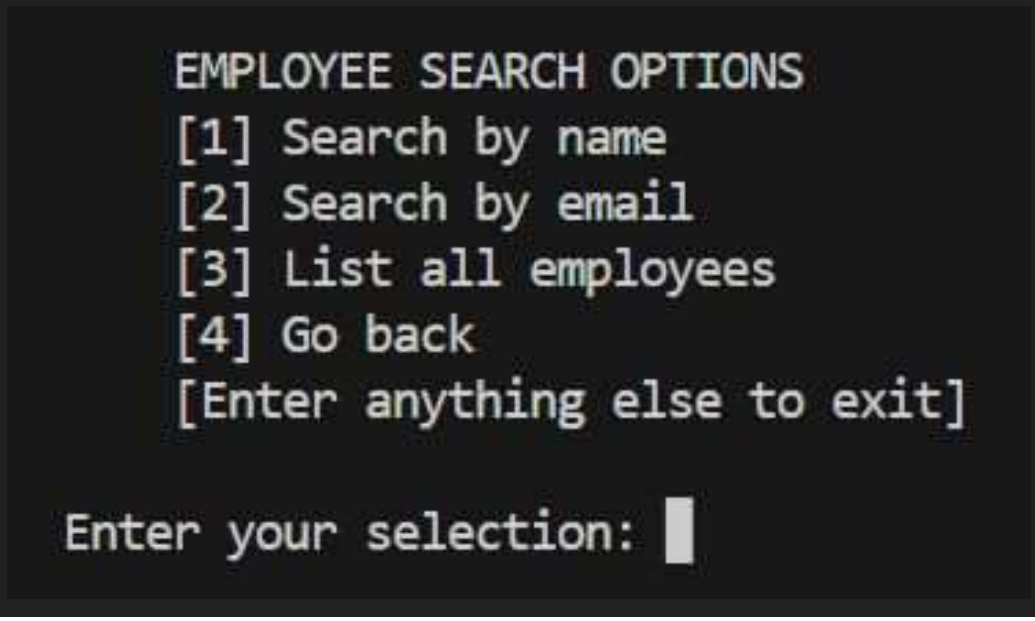
Theo Marentes

Human Resources Manager



Application Features

Search Employees



Export Search To CSV

```
Dwight Schrute, dwight.schrute@gmail.com, sales, 87000
Angela Martin, angela.martin@gmail.com, accounting, 72000
Meredith Palmer, angela.martin@gmail.com, relations, 74000


Type 'Yes' to export as csv, enter anything else to skip: yes
CSV Created 'export.csv'
```

Edit Employee

```
|   EMPLOYEE EDIT OPTIONS
|   [1] Find by name
|   [2] Find by email
|   [3] Go back
|   [Enter anything else to exit]

Enter your selection:
```

Add Employee

```
Enter a name: Michael Scott
Enter an email: michael.scott@gmail.com
Enter a department: manager
Enter a salary: 105000
```

Delete Employee

```
|   EMPLOYEE DELETE OPTIONS
|   [1] Delete by name
|   [2] Delete by email
|   [3] Go back
|   [Enter anything else to exit]

Enter your selection:
```

Code Overview

Interface Navigation

```
match option:
    case 1:  # search employees···

    case 2:  # edit employee···

    case 3:  # new employee···

    case 4:  # delete employee···
```

Interface Navigation

```
try:
    option = UserInput.main_menu_selection()
except ValueError:
    print("Closing application")
    break
```

```
def main_menu_selection() -> int:
    selection = int(input("Enter your selection: "))
    if selection in (1, 2, 3, 4):
        return selection
    else:
        raise ValueError
```

```
match option:
    case 1:  # search employees···

    case 2:  # edit employee···

    case 3:  # new employee···

    case 4:  # delete employee···
```

List All Employees

```
def list_all() -> list:
    print("NAME, EMAIL, DEPARTMENT, SALARY")
    employee_list = []
    for i in range(len(Employee.employees)):
        line = ""
        line = Employee.employees[i].name+", "+Employee.employees[i].email+", " + \
            Employee.employees[i].department+", " + \
            str(Employee.employees[i].salary)
        employee_list.append([Employee.employees[i].name, Employee.employees[i].email,
                              Employee.employees[i].department, str(Employee.employees[i].salary)])
        print(line)
    return employee_list
```

Export Search To CSV

```python
def export_search_csv(data) -> None:
    with open('export.csv', 'w', newline='') as f:
        writer = csv.writer(f)
        writer.writerow(["Name", "Email", "Department", "Salary"])
        for i in range(len(data)):
            list_data = []
            for j in range(0,4):
                list_data.append(data[i][j])
            writer.writerow(list_data)
    print("CSV Created 'export.csv'")
```

HTTP Request

```python
def get_quote():
    data = requests.get('https://api.api-ninjas.com/v1/quotes?
    category=inspirational', headers={'X-Api-Key': 'r/
    EG6kiJC3Rc5V7pwcXsWQ==321MFqColfB9oTck'}).json()
    return(data)
```

Time Module

```
local = time.localtime()
local_hours = local.tm_hour
local_mins = local.tm_min
local_year = local.tm_year
local_month = local.tm_mon
local_day = local.tm_mday

Time: {local_hours}:{local_mins} - Date: {local_day}/{local_month}/{local_year}
```

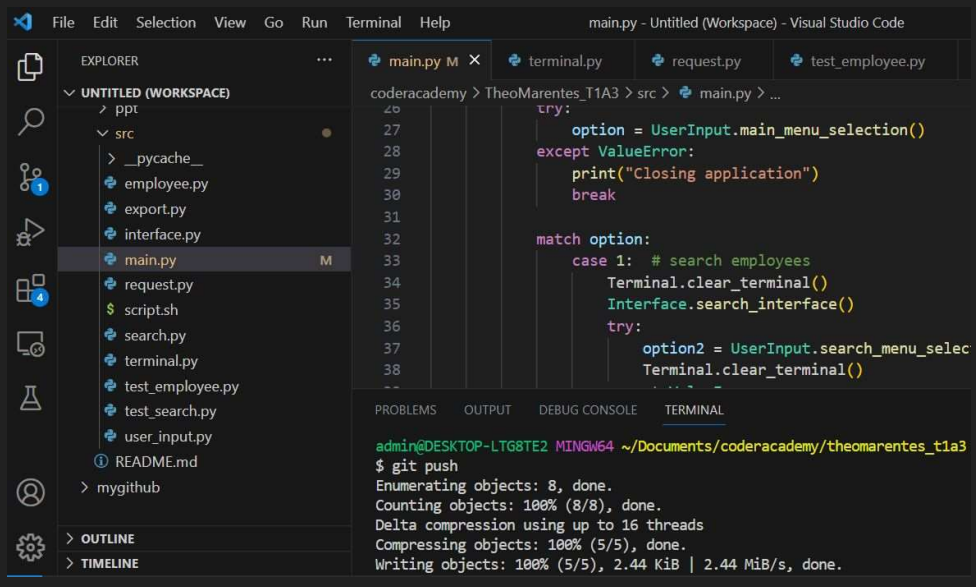
Development Process

Initial Steps

1. Create trello board
2. Create github repository
3. Connect git
4. Create initial files

Development Environment

- VSCode 1.79.2
- Python 3.10.5
- Git 2.40.1
- Pip 22.0.4



Development Challenges

```
for i in range(len(data)):
    list_data = []
    for j in range(0,4):
        list_data.append(data[i][j])
    writer.writerow(list_data)
```

Development Challenges

```
if python -c "import requests" &> /dev/null; then
    echo 'Requests is already installed.'
else
    echo 'Requests is not installed. Type "Y" to install rec
    read user_input
    if [ "$user_input" == "Y" ] || [ "$user_input" == "y" ]
    then
      pip install requests
    fi
fi
```

Development Challenges

```
def get_quote():
    data = requests.get('https://api.api-ninjas.com/v1/
    quotes?category=inspirational', headers={'X-Api-Key': 'r/
    EG6kiJC3Rc5V7pwcXsWQ==321MFqColfB9oTck'}).json()
    return(data)
```

Ethical Issues

Privacy and Data Protection

Informed Consent

Accessibility and Inclusivity

Favourite Part

Coding

```
match option:
    case 1:  # search employees
        Terminal.clear_terminal()
        Interface.search_interface()
        try:
            option2 = UserInput.search_menu_selection()
            Terminal.clear_terminal()
        except ValueError:
            print("Closing application")
            break
```

T1A3: Terminal Application

Theo Marentes