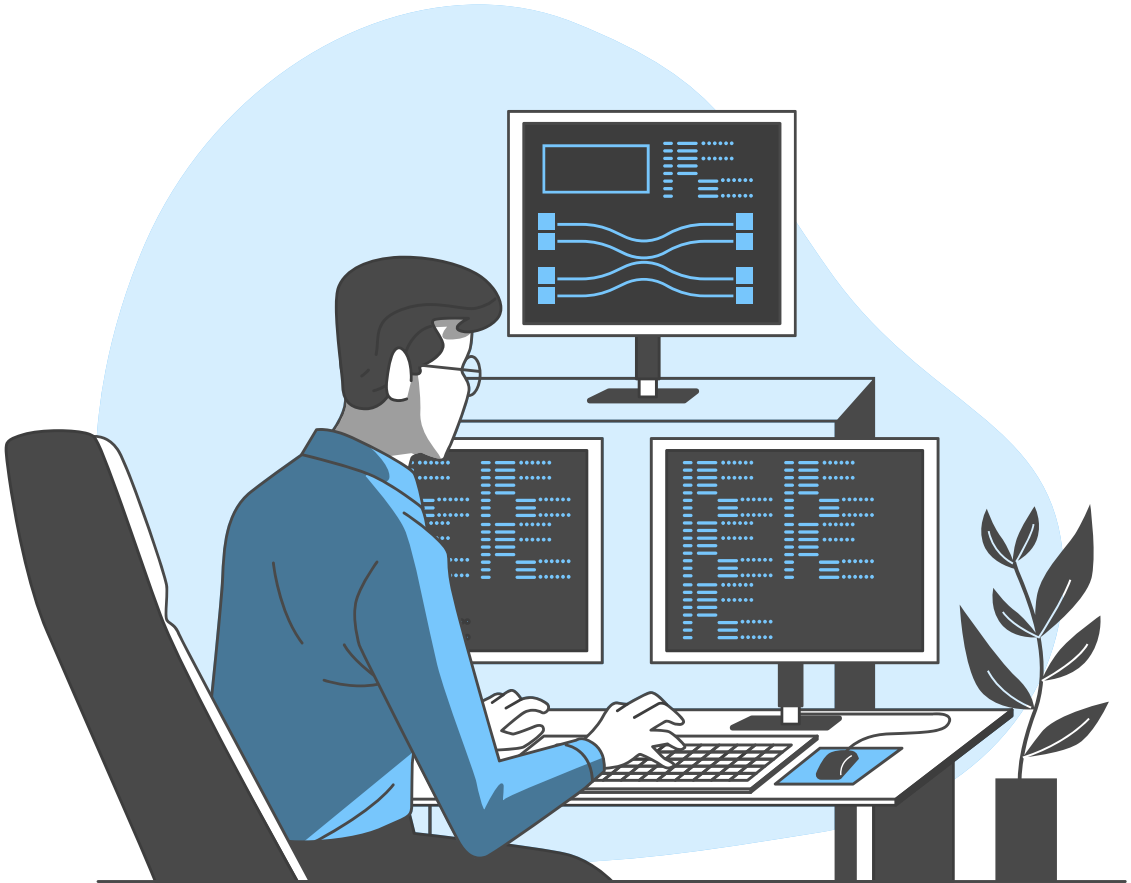
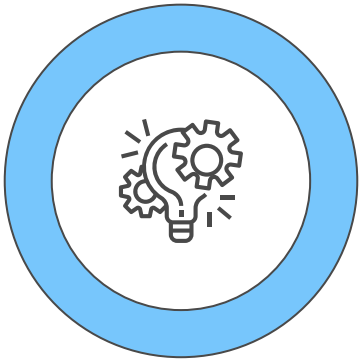


DEFECT PROGRAMMER ASSIGNMENT

Capgemini - Sprint 1
Group 3



THE PROJECT IDEA

To develop a software that automatically assigns the defects reported by the client company to programmers depending on the functional area they are handling.

The Need ?

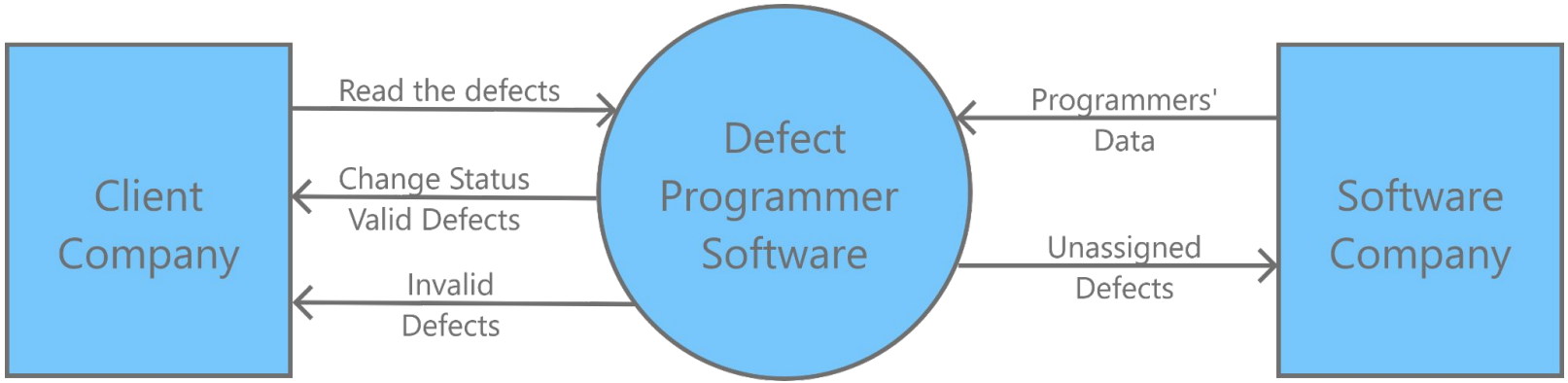
01 Time Consuming

02 Not realtime

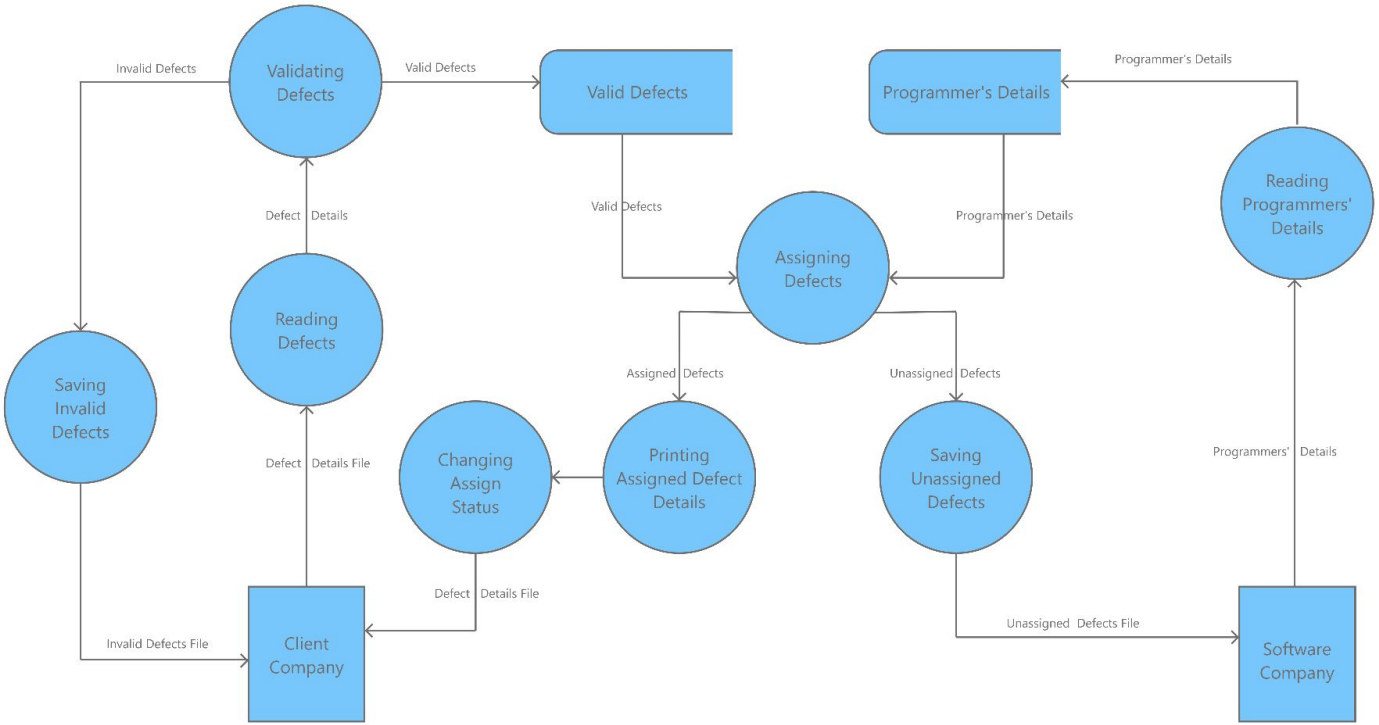
03 More Labour

04 Less Efficient

Our Solution



DFD 1


Validating Defects
Invalid Defects
Valid Defects
Valid Defects
Programmer's Details
Programmer's Details
Reading Programmers' Details
Defect Details
Valid Defects
Programmer's Details
Assigning Defects
Reading Defects
Saving Invalid Defects
Assigned Defects
Unassigned Defects
Defect Details File
Changing Assign Status
Printing Assigned Defect Details
Saving Unassigned Defects
Programmers' Details
Defect Details File
Invalid Defects File
Client Company
Unassigned Defects File
Software Company

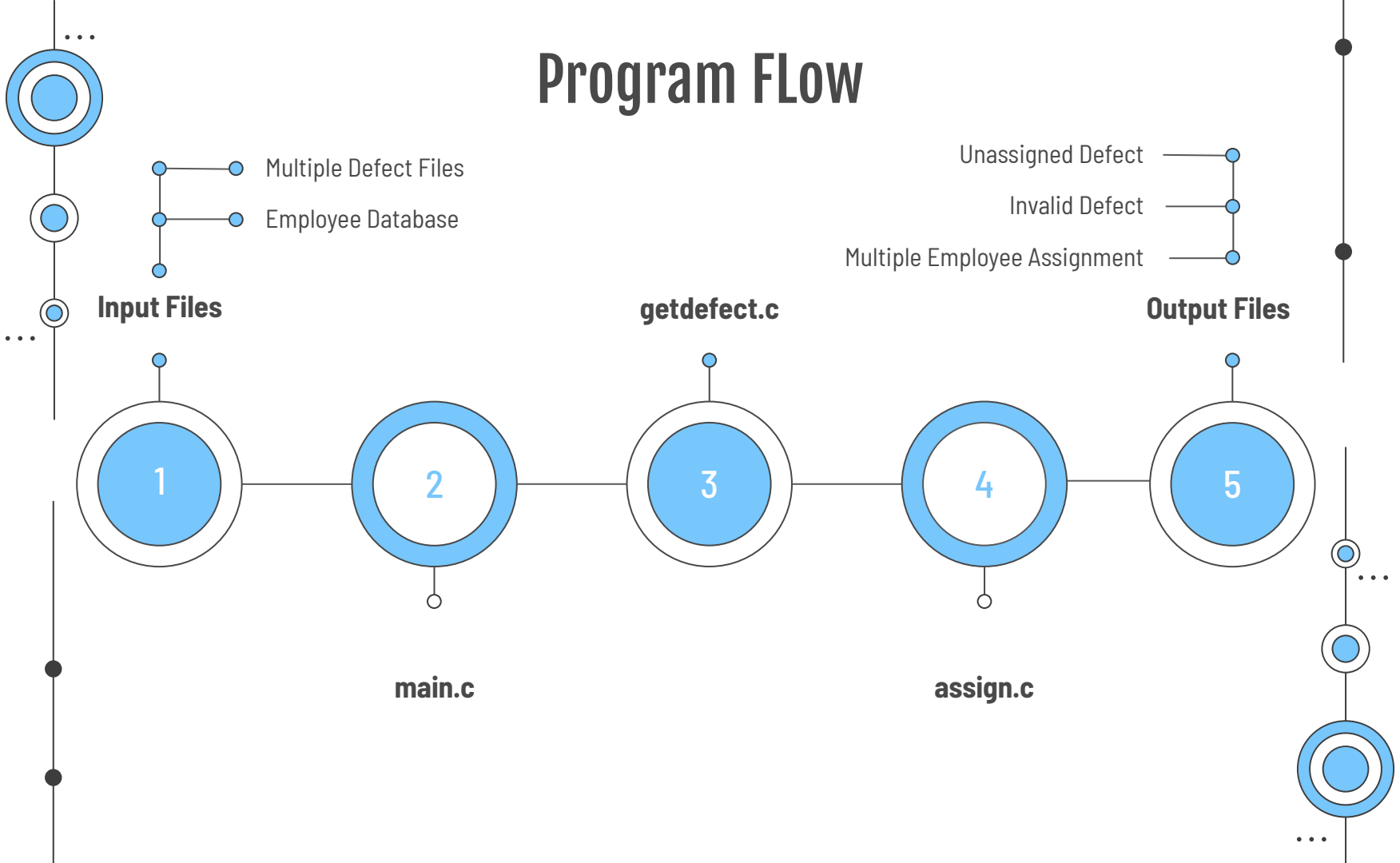
Our Solution is

Multi Threaded

Multi File & Multi Directory System

Modular

Program FLow



01

main.c

STRUCTURES

01

Defect

```
Struct defect{
Char *defectID;
Char *description;
Char *moduleName;
Char *functionalArea;
Char *date;
Char *status;
Char *type;
};
```

02

Employee

```
Struct employee{
Char *Id;
Char *Name;
Char *BUnit;
Char *Expertise;
Char *Designation;
Pthread_mutex_t emplock;
Int n_defect;
Defect *assigned_arr[MAX];
};
```

1. main()

Input Defect files are taken as command line arguments and also validates them.

Separate threads are created for each input files and these files are passed to getDefect() Function.

It calls getEmployee() function to fetch data from Employee Database.

Finally it waits for all threads to complete their work.

2. getEmployee()

It opens “employee.txt” database file.

Now it reads the file line by line, each line contains information of one employee.

It stores this information inside Employee Structure.

Displays error if file can't be opened for any reason.

02

getdefect.c

Functions

1.Get Defect

Reads defects from the input and and calls checkvalidity and call assignEmployee for valid defect

2.Check Validity

Returns true if count is equal to 7 and else return false

3.Valid defect

It stores the valid defect in Defect structure

4.Invalid Defect

Display invalid defect message and append it into invalidDefect.txt

1. GetDefect()

It reads defects from the input file

calls checkvalidity if true call validDefect()
Else call invaliddefect()

It calls assignEmployee() for valid defect
.

2. CheckValidity()

We initialize count = 0;
It divides the string into token using strtok
And increment count for each attribute

: is delimiter
char *token = strtok(s, ":");

If count is equal to 7 , it returns true
Else returns false

.

3. ValidDefect()

It tokenizes the string using strtok with
And dynamically allocates memory and
stores them into their respective
Attribute in defect structure.

.

4. InvalidDefect()

It display invalid defect message with defect id

and appends it into invalidDefect.txt

.

03

assign.c

Functions

01

assignEmployee()

Checks for defects with status as open.

02

searchProgrammer()

Searches for programmer suitable of open defect.

03

unassignedDefect()

Copies all unassigned Defect into separate text file.

04

createEmployeeFile()

Creates separate files for each programmer who have at least one defect assigned to him

1. assignEmployee()

It loops through all defects and checks their status.

If status is open then it calls searchProgrammer() Function.

Defects with any other status are ignored.

2. searchProgrammer()

Now for each defect passed, it searches for programmer in the array.

Search Criteria:
Functional Area of Defect shall match with **Expertise** of Programmer.

If there is a match, the defect is assigned to the programmer and createEmployeeFile() Function is called.

If no programmer could be found then uassignedDefect() Function is called for that defect.

MUTEX SNAPSHOT

```
if (strcmp(defectptr->functionalArea, arr[i]->Expertise) == 0)
        {
            foundflag = 1;
            defectptr->status = "Assigned";
            pthread_mutex_lock(&arr[i]->emplock);
            arr[i]->n_defect++;
            arr[i]->assigned_arr[(arr[i]->n_defect) - 1] = defectptr;
            createEmployeeFile(arr[i], defectptr);
            pthread_mutex_unlock(&arr[i]->emplock);
            break;
        }
```

3. unassignedDefect()

Now it opens “uassignedDefect.txt” file and appends all information of current defect to the last line of the file.

If file is not present it creates a new one.

Displays proper error if there is any issue with opening or writing inside this file.

4. createEmployeeFile()

Creates separate file for each employee, if not present already, who have at least one defect assigned to them.

Filename: **<EmpID>_assignments.txt**

Appends employee and defect information into the file.

Displays proper error if there is any issue with opening or closing of employee file.

```
sprintf(out_file, "../data/out/%s_assignments.txt", emp_ptr->Id);
```

Testing

Unit testing and Integration testing

Files

Defect.txt

defectID : description : moduleName : **Functional Area** : date : status : type

1. F001:Column values in BOM reports are incorrect:Aircraft design:BOM reports:21/08/2022:open:fatal
2. F002:Unit prices are not shown while preparing invoice:Invoices:Display products:23/04/2022:close:fatal
3. N001:BOM report columns not aligned properly:Aircraft design:BOM reports:21/08/2022:open:niceToHave
4. 0001:Aircraft:BOM reports:21/08/2022:open:niceToHave
5. F003:Column values in client dashboard not shown:Aircraft design:Manage customers:21/08/2022:open:fatal

Employee.txt

employeeID : Name : Business Unit : **Expertise** : Designation

1. A123:Suresh Panchal:UK Telecom:BOM report:Principal engineer
2. D012:J K Laxmi:Finacle Systems:Display products:Junior programmer
3. C015:Sandeep Khaire:UK Telecom:Manage customers:Senior programmer
4. D002:Mahesh Katkar:Pharmaceutical Systems:Licensing:Principal engineer
5. B011:Sreehari Bhaskar:DBMS Department:Data manager:Senior Analyst

Different Types of Test Cases Covered

- Less than or More than Actual Defect Attributes
- Less than or More than Actual Employee Attributes
- No Programmer is Found for Defect
- More than one Programmer is Found
- File is Empty or not Opening or Invalid file type
- Wrong Format of Defects or Employee in File
- Multiple Defect.txt Files
- Defect Status Close/Open Or Something else
- Checking for uppercase and lowercase
- Defect With same Defect IDs
- Date Format is Wrong
- Space ' ' is Considered as Character EX:- : BOM report :--

TEST SUITE

Sunny Test Cases

1. "F001:Column values in BOM reports are incorrect:Aircraft design:BOM report:21/08/2022:open:fatal";
2. "N001:BOM report columns not aligned properly:Aircraft design:BOM report:21/08/2022:open:niceToHave";
3. "F002:Unit prices are not shown while preparing invoice:Invoices:Display products:23/04/2022:close:fatal";

Rainy Test Cases

1. "ID01: : : ::open:";
2. "O001:Aircraft:BOM reports:21/08/2022:open:niceToHave";
3. "M001:Sed ut perspiciatis unde omnis iste:Aircraft design:consequatur:20/08/2002:open:lagging:lagging";

Unit Testing for checkValidity function

```
CUnit - A unit testing framework for C - Version 2.1-3
http://cunit.sourceforge.net/

Suite: Basic_Test_Suite1
  Test: Testing Sunny Cases ...passed
  Test: Testing Rainy Cases ...passed

Run Summary:    Type  Total    Ran   Passed  Failed Inactive
                suites    1      1     n/a      0        0
                 tests    2      2       2      0        0
               asserts   10     10      10      0      n/a

Elapsed time =    0.000 seconds
```

Check Validity()

It takes one argument, that is :

1. String pointer

It divides the string into token using strtok
And increment count for each.

If count is equal to 7 , it returns true
Else returns false

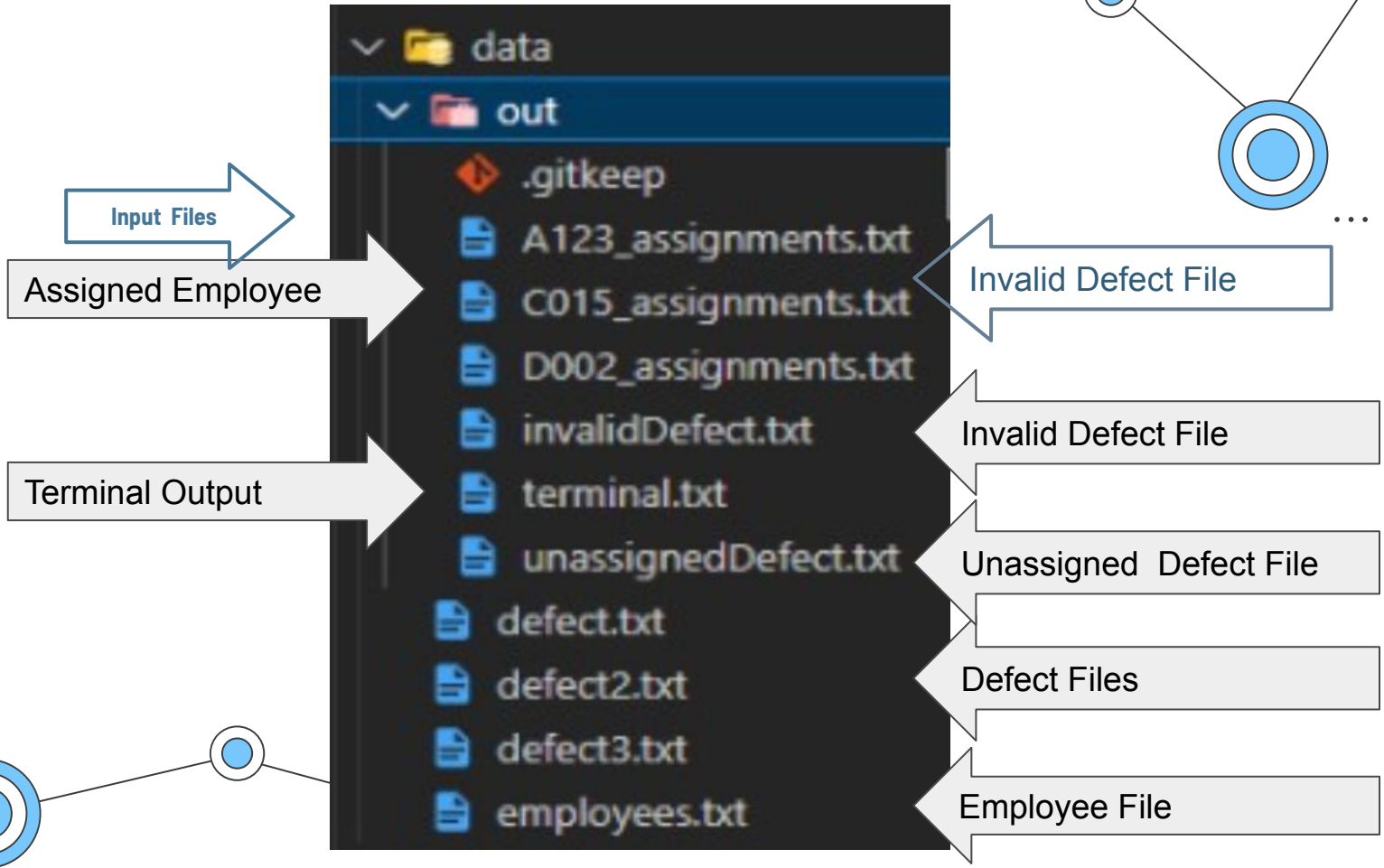
.

INTEGRATION TESTING FILES DIRECTORIES

```
$ ls -lh
total 8.0K
-rwxrwx---+ 1 Karan Karan 466 Sep 14 21:24 defect.txt
-rwxrwx---+ 1 Karan Karan 495 Sep 14 21:24 defect2.txt
-rwxrwx---+ 1 Karan Karan 396 Sep 14 21:25 defect3.txt
-rwxrwx---+ 1 Karan Karan 470 Sep 13 13:11 employees.txt
drwxrwx---+ 1 Karan Karan   0 Sep 14 21:32 out
```


```
total 9.0K
-rwxrw-r--+ 1 Karan Karan  513 Sep 14 21:32 A123_assignments.txt
-rwxrw-r--+ 1 Karan Karan  265 Sep 14 21:32 C015_assignments.txt
-rwxrw-r--+ 1 Karan Karan  234 Sep 14 21:32 D002_assignments.txt
-rwxrw-r--+ 1 Karan Karan  175 Sep 14 21:32 invalidDefect.txt
-rwxrw-r--+ 1 Karan Karan 3.0K Sep 14 21:32 terminal.txt
-rwxrw-r--+ 1 Karan Karan  222 Sep 14 21:32 unassignedDefect.txt
```

Integration Testing Output



Integration Testing - Terminal output

```
--- Total files in queue: 3 —
```
1

```
--- Total Employee: 7 ---

ID: A123 Name: Suresh Panchal
ID: D012 Name: Karan Karan
ID: C015 Name: Shreehari Khaire
ID: D002 Name: Mahesh Katkar
```
2

```
--- Creating Thread for file 1: /data/defect.txt ---
--- Creating Thread for file 2: /data/defect2.txt —
--- Processing file: ../data/defect.txt
Defect ID: O001 contains insufficient
information.
--- Creating Thread for file 3: /data/defect3.txt —
--- Processing file: ../data/defect2.txt
Defect ID: M001 contains insufficient
information.
--- Cannot open file: ../data/defect3.txt
```
3

```
--- Total Valid Defects: 4 ---

ID: F001 Status: open
ID: F002 Status: close
ID: N004 Status: open
ID: K002 Status: open
```
4

PTO

```
--- Searching Programmer for defect Id: F001 ---
--- Searching Programmer for defect Id: N004 ---

Defect Id: F001
Status: Assigned
Module Name: Aircraft design
Functional Area: BOM report
Description: Column values in BOM reports are
incorrect

Has been assigned to:-
Employee Id: A123
Employee Name: Suresh Panchal
```
5

```
--- Searching Programmer for defect Id: N001 ---
Defect Id: N004
Status: Assigned
Module Name: Aircraft design
Functional Area: BOM report
Description: BOM report columns not aligned
properly

Has been assigned to:-
Employee Id: A123
Employee Name: Suresh Panchal
```

PTO

```
--- Searching Programmer for defect Id: F003 ---
--- Searching Programmer for defect Id: K002 ---

Defect Id: F003
Status: Assigned
Module Name: Aircraft design
Functional Area: Manage customers
Description: Column values in client dashboard
are not shown

Has been assigned to:-
Employee Id: C015
Employee Name: Shreehari Khaire

Defect Id: K002
Status: Assigned
Module Name: reprehenderit
Functional Area: BOM report
Description: sed quia non numquam eius modi
tempora incidunt

Has been assigned to:-
Employee Id: A123
Employee Name: Suresh Panchal
```

```
--- Searching Programmer for defect Id: K003 ---
--- Programmer not found for defect Id: K003 ---
```
6

TEAM CG81-GROUP 3

- Aman Bhaskar
- Karan Nareshbhai Telar
- Sreehari P S
- Singuluri Sai Vamsee
- Krishna Chaitanya Chekka

THANKS!

Do you have any questions?

Scan below code or visit:
https://github.com/sreeharipavvatta/CGSprint1