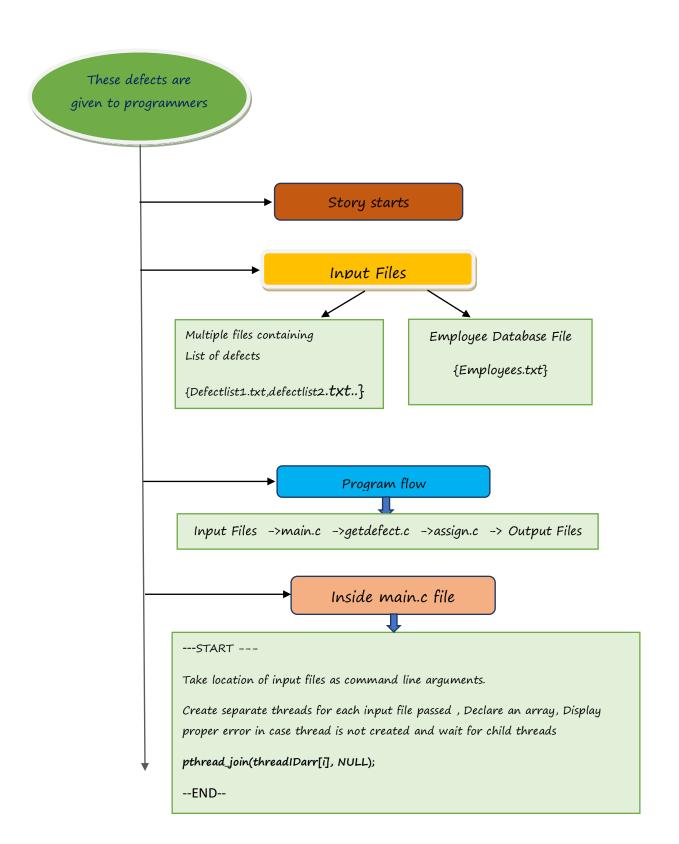
PROJECT SPRINT1 GROUP-3

STORY: The story is about describing that the clients of software development company report defects in software they purchased.



Inside getdefect.c file



Take the location of file as input, open file with file pointer

If pointer is NULL, Print "cannot open file" and exit

If line returns NULL return as we reached the end of file

Else

Pass the checkValidity function

--END--

Function Checkvalidity(char *str):

Takes defect {string} as input, checks for validity. If any information is not present it is invalid entry.

Call handleInvalidityDefect() for each invalid entry. Pass defectID as argument

Call handle Valid Defect() for each valid entry. [Pass]

Function Invalid Defect (char * DefectID)

--START--

Takes Invalid Defect ID and Defect String/Line as Input

PRINT "Unavalible Info of < Defect ID>

IF file is not created

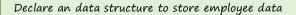
CREATE "invaliddefect.txt"

APPEND entire invalid defect line

--ELSE --

APPEND entire invalid defect line

Inside assign.c file



Create an array to store employee data. Call function getEmployee and pass this array. Checks for status of all elements inside array.

IF status is not open ignore it

IF status is open then call seachProgrammer Function,

pass array of employee structure and array of valid defect strucutre to it.

Function getEmployee

Takes array of employee structure as an argument Open input file with employee data using File pointer Display error if file can't be opened for any reason

Store these information in employee structure

Store all employee in the array

Function searchProgrammer

Takes array of employee structure and array of valid defect entry structure as argument. IF multiple programmer is found assign to first one.

After assigning change defect status to assigned. Call createEmployeeFlle Function, pass Employee and defect sturcuture.IF programmer is not found, call unsignedDefect Function, pass defect structure to it.

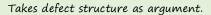


Takes array of valid defects entry as argument

Checks for status = assigned.

Defects assigned to programmers to be displayed along with defect description, module name, functional area, filed-on date type, Emp ID and EMP Name.

Function unassigned Defect



Appends information of defect into a file unassigned Defects.txt

IF file doesn't exist - create new file with same name

Function create Employee File

Takes structure of Employee and Defect

If employee file is exists, append defect information that has been assigned in that file

Else

Create file with filename <empID>_assignemnts.txt and copy defect information.

FINISHED