

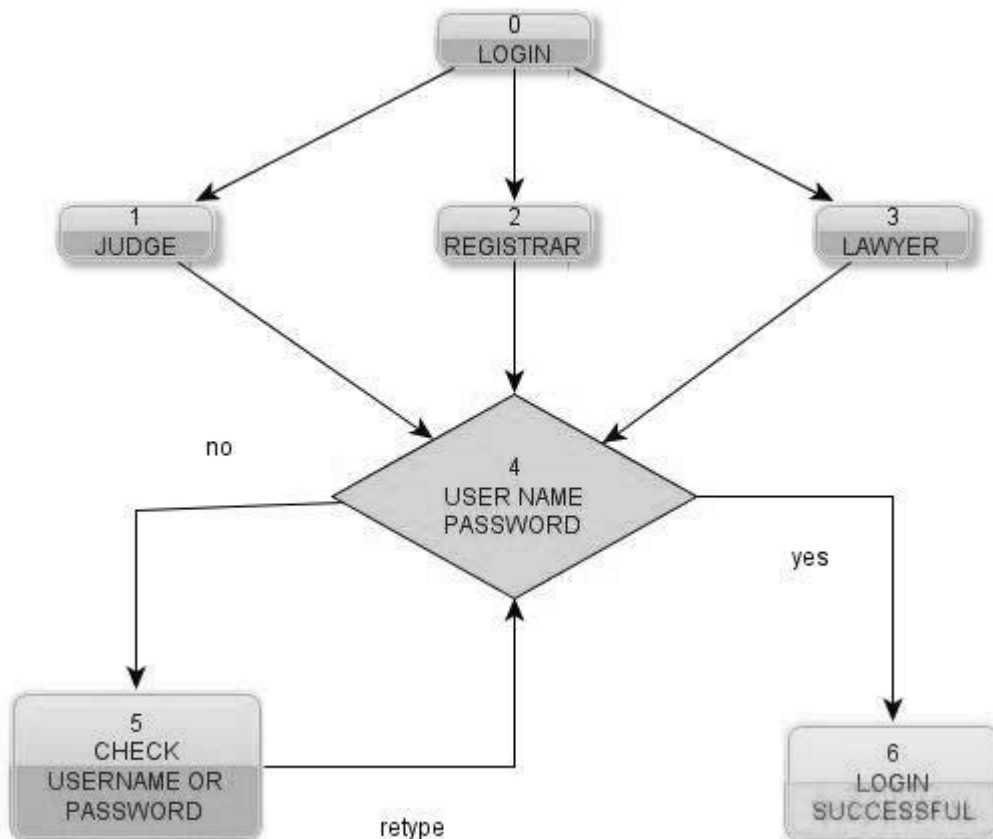
WHITE BOX TESTING

- Group no 35

White box testing is the way of testing the software that tests the internal workings of an application, as opposed to its functionality (i.e. black box testing). We use test cases to exercise all the paths through the code. The test cases must be in such way that all the possible cases are tested.

We here are testing all possible cases with flow diagrams shown for all possible cases :

LOGIN



PARAMETERS :

String : Username , Password

Combo Box Selector : Registrar / Judge / Lawyer

Data Members :

Registrar : It has saved data of Registrar login and password details.

JudgeDataBase : It has the details of Judge in the form <ArrayList> which is traversed to check for a judge.

LawyerDataBase : It has the details of Lawyer in the form <ArrayList> which is traversed to check for a lawyer.

Test Cases :

We have 6 independent paths here.

1. 0-2-4-6 Registrar logs in successfully as the user name and password matches with those stored.
2. 0-2-4-5-4 Registrar enters wrong Password and which doesn't match with those stored so he is taken back to login screen.
3. 0-1-4-6 Judge logs in successfully as the user name and password matches with those stored.
4. 0-1-4-5-4 Judge enters wrong Password and which doesn't match with those stored so he is taken back to login screen. It may be due to the mismatch of judge username or password.
5. 0-3-4-6 Lawyer logs in successfully as the user name and password matches with those stored.
6. 0-1-4-5-4 Lawyer enters wrong Password and which doesn't match with those stored so he is taken back to login screen. It may be due to the mismatch of lawyer username or password.

SCHEDULE FOR REGISTRAR :

Parameters : **Button Click**

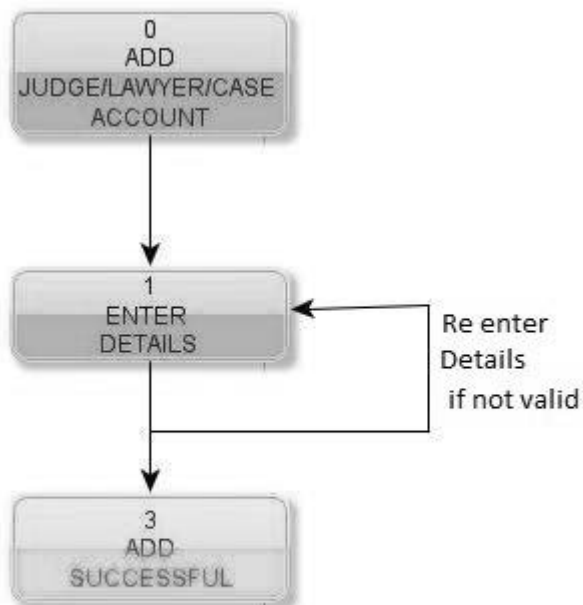
Data Members :

Slot Data Base :All the slots are checked and those which match the date with the present are shown.

Test Cases : Only 1 test case.

Click shows you the any day schedule .

ADD NEW ACCOUNT



Parameters :

String: Add respective fields like(name, address, mobile no.) for new Judge / lawyer accounts and necessary case details (case type , defendant , case date .. etc) for new Case.

Data Members :

CaseDataBase : To add new case.

LawyerDataBase : To add new Lawyer account.

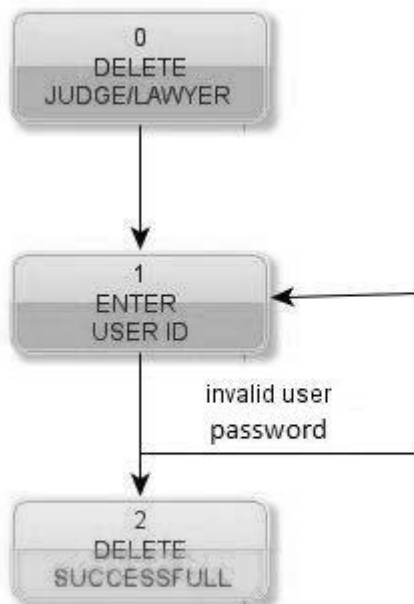
JudgeDataBase : To add new Judge account.

TestCases :

There are 2 paths possible. Here as all the accounts and the type of errors are similar all are included into same flow chart.

- 1 0-1- 3 When all details are fair new account is added to respective database, Case/Lawyer/Judge
- 2 0-1-1 When there is error in format like date(dd/mm/yyyy) or any important field is empty then the error is shown and is taken back to the Add Account frame.

DELETE JUDGE/LAWYER ACCOUNTS



Parameters :

String : Password (Judge/Lawyer)

Data Members :

LawyerDataBase : To check required Lawyer password .

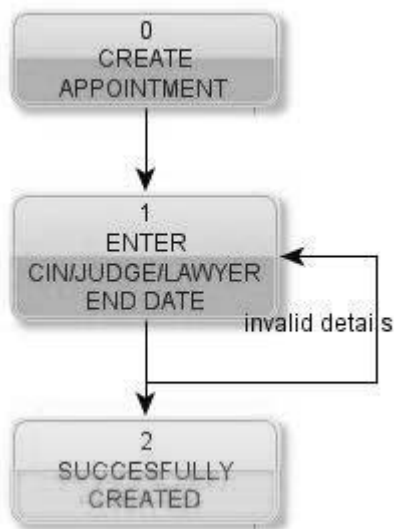
JudgeDataBase : To check required Judge password .

Test Cases :

Only 2 cases are possible. Lawyer or Judge accounts are similar in deletion.

- 1 0-1-Respective login user password is matched and deleted from database
- 2 0-1-Respective login user password is not found, error message is shown and taken back to enter again.

CREATE HEARING



Parameters :

String : Enter CIN, select Judge/Lawyer , enter expected date

Data Members :

SlotDataBase :Case is allotted a slot and case is added to SlotDataBase .

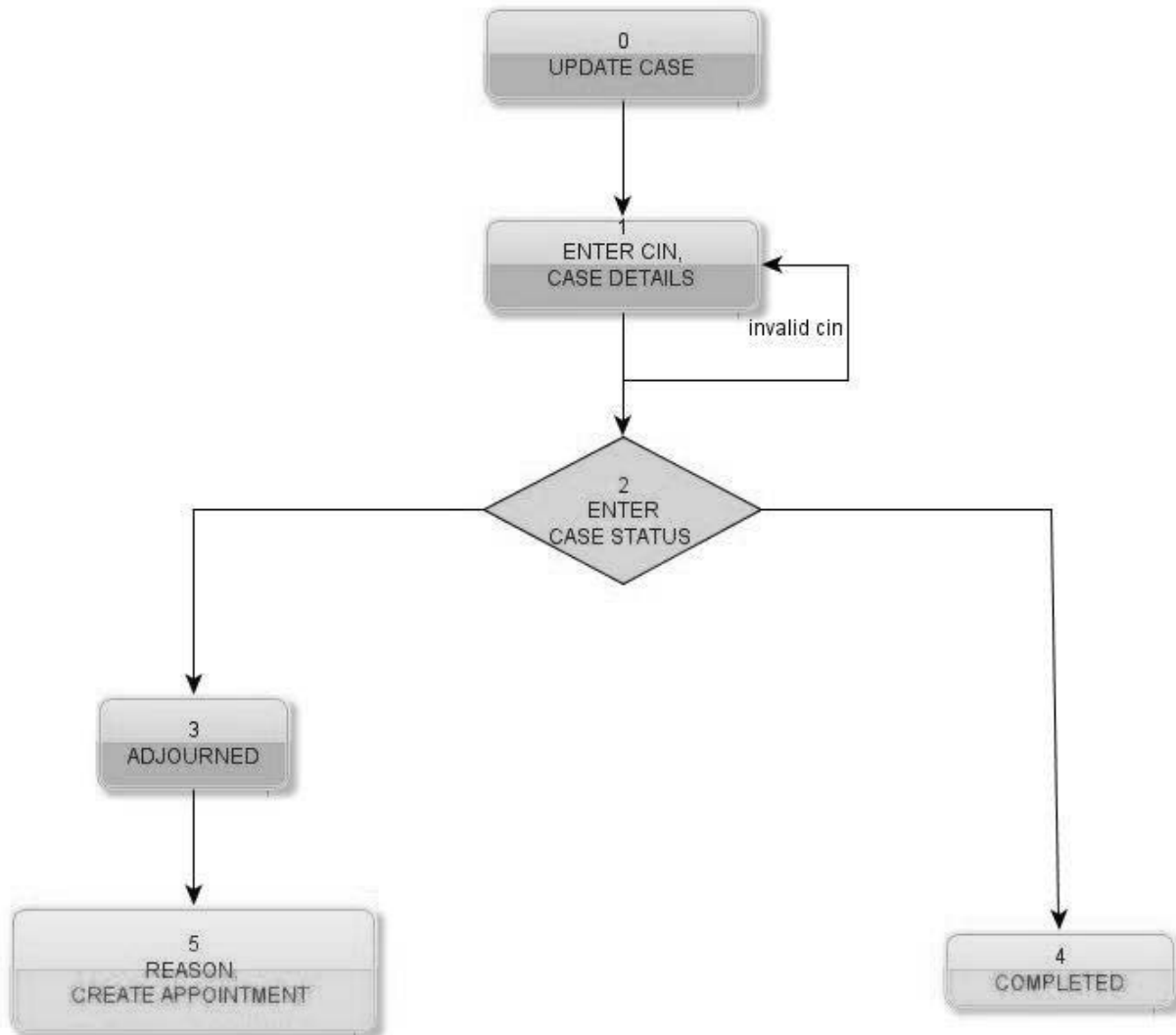
CaseDataBase :Case is searched by CIN and case is updated in CaseDataBase.

TestCases :

There are 2 test cases possible. Appointment is created.

1. 0-1-2 All details are proper and the case is given an appointment.
2. 0-1-1 if details are having any format mis matches then it is taken back to the Appointment frame.

UPDATE CASE :



Parameters :

String : Enter CIN, case status, Case summary is also entered. If case is adjourned then reason for adjournment is also added. If case is completed then judgement is also added.

Data Members :

CaseDataBase : Case is traversed through this file and case is updated.

TestCases :

There are 3 test cases.

1. 0-1-1 If the CIN entered is not valid , it is taken back.
2. 0-1-2-4If the case status is updated to is Completed.
3. 0-1-2-3-5 If the case is Adjourned , then the reason for Adjournment is taken and new Appointment is taken.

CASE QUERY

Parameters :

String : CIN is entered.

Data Members :

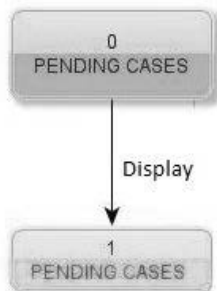
CaseDataBase : CIN is traversed through Case Data Base and Case status is shown.

Test Cases :

There are 2 possible test cases.

1. Invalid CIN , if CIN is not found in CaseDataBase , it is returned.
2. When CIN is valid ,the case status is shown whether it is present or has taken appointment or is adjourned or is completed and some other details of cases are also shown.

PENDING CASES

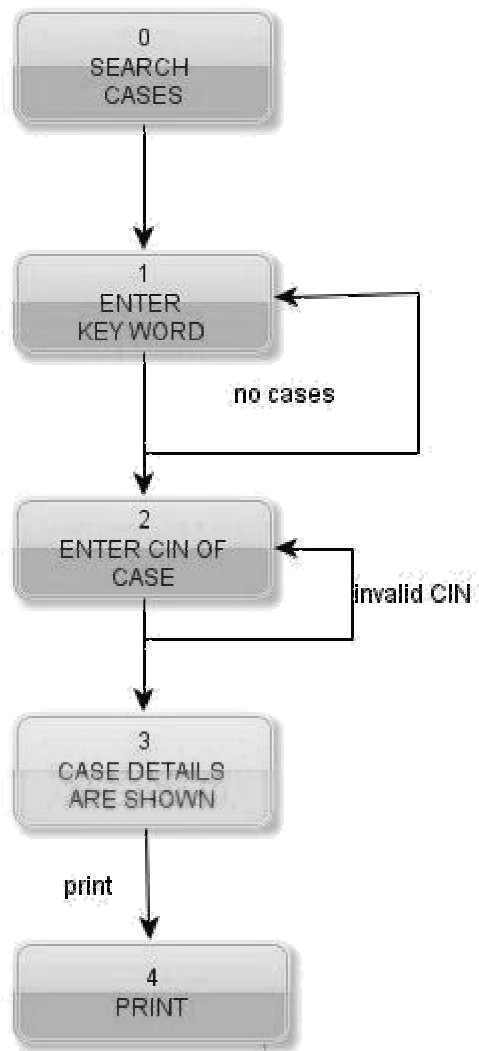


Parameters : Click the Pending Cases

Data Members : Pending Case Data Base All the cases are traversed through Pending Case Data Base.

Test Cases : Checking Pending Case

CASE SEARCH (JUDGE)



Parameters :

String : Enter any key word related to the case to be enquired.
 Enter CIN from the cases shown from related search.

Data Members :

CaseDataBase : Cases are traversed for case search for key word.

Test Cases :

There are 4 possible test cases.

- 1 0-1-1 , If there are no cases related to search then he has to search again
- 2 0-1-2-2 if he copies wrong CIN from table no results are shown
- 3 0-1-2-3 Case details are shown
- 4 0-1-2-3-4 Case details can be printed.

CASE SEARCH LAWYER

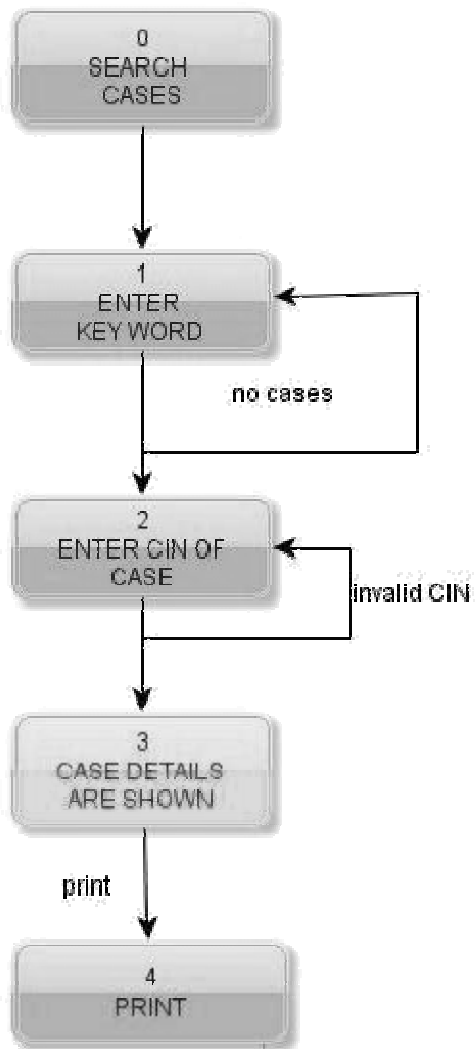
Parameters :

String : Enter any key word related to the case to be enquired.

Enter CIN from the cases shown from related search.

Data Members :

CaseDataBase : Cases are traversed for case search for key word.



Test Cases :

There are 4 possible test cases.

1 0-1-1 , If there are no cases related to search then he has to search again

2 0-1-2-2 if he copies wrong CIN from table no results are shown

3 0-1-2-3-4 Case details are shown

Actual Testing Done

In this section, we describe the software development process model used and the testing that has been performed in different stages of the product development. It includes all the three types of testing mentioned in the earlier section.

Process Model

In the creation of JIS, we used the popular and simple Iterative Development model. We justify the selection of this model as follows:

- The problem was well understood.
- The project was known to be of short duration, and time was of essence.
- New requirements were realized on further developing the software, and incremental addition of features was of utmost importance.
- This model allowed the prioritizing of requirements, as well as quick deliveries of versions.
- This model also allowed the incorporation of user feedback, and appropriate changes.

Unit Testing

The unit testing has been implemented using the white box test cases described in Section 4. For unit testing, we wanted to write test cases such that each of these paths is tested at least once. We successfully modeled these test cases in Section 4.

System Testing

Ideally, we would like to test every possible thing that can be done with our program. But, as we said, writing and executing test cases is expensive. We wanted to make sure that we definitely write test cases for the kinds of things that the customer will do most often or even fairly often. Our objective was to find as many defects as possible in as few test cases as possible. We started by looking at each customer requirement to make sure that every single customer requirement has been tested at least once.

Regression Testing

We have used the above described Unit Tests and Black Box Tests to evaluate our system at various stages of development. Whenever a new iteration of our software was generated, we tested it on the applicable subsets of the above described tests, and proceeded to add more features only after the bugs discovered were eliminated. So, in effect we have implemented Regression Testing.