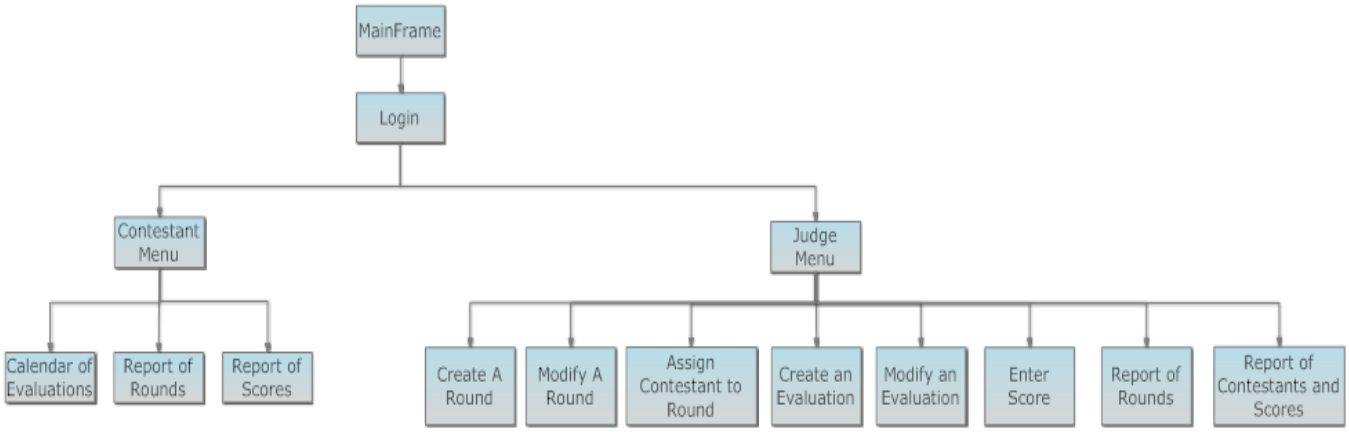
Incremental and Regression Testing

➢ Classification of Components

- Components structure



- Detail of each component
  - *MainFrame*
    - Input: None
    - Output: None
    - Parent Dependency: None
    - Child Dependency: Login
  - *Login*
    - Input: User ID; Password
    - Output: Credential validation (success / failure)
    - Parent Dependency: MainFrame
    - Child Dependency: ContestantMenu, JudgeMenu
  - *JudgeMenu*
    - Input: None
    - Output: None
    - Parent Dependency: Login
    - Child Dependency: JudgeCreateARound, JudgeModifyARound, JudgeAssignContestantToRound, JudgeCreateEvaluation, JudgeModifyEvaluation, JudgeEnterScore, JudgeReportOfRounds, JudgeReportOfContestantsAndScores
  - *JudgeCreateARound*

        - Input: Name; Season; Year; Time; Location
        - Output: Database operation status (success / failure)
        - Parent Dependency: JudgeMenu
        - Child Dependency: None
    - *JudgeModifyARound*
        - Input: Selected Round; Name; Period; Year; Meets_At; TestRoom
        - Output: Database operation status (success / failure)
        - Parent Dependency: JudgeMenu
        - Child Dependency: None
    - *JudgeAssignContestantToRound*
        - Input: Contestant ID; Round Name
        - Output: Database operation status (success / failure)
        - Parent Dependency: JudgeMenu
        - Child Dependency: None
    - *JudgeCreateEvaluation*
        - Input: Round; Evaluation Type; Percent Weight; Deadline
        - Output: Database operation status (success / failure)
        - Parent Dependency: JudgeMenu
        - Child Dependency: None
    - *JudgeModifyEvaluation*
        - Input: Round; Evaluation Type; Percent Weight; Deadline
        - Output: Database operation status (success / failure)
        - Parent Dependency: JudgeMenu
        - Child Dependency: None
    - *JudgeEnterScore*
        - Input: Round; Evaluation; Contestant; Score
        - Output: Database operation status (success / failure)
        - Parent Dependency: JudgeMenu
        - Child Dependency: None
    - *JudgeReportOfRounds*
        - Input: None
        - Output: Round Name; Time; Location; # of Contestants; # of evaluations; # of Notices
        - Parent Dependency: JudgeMenu
        - Child Dependency: None
    - *JudgeReportOfContestantsAndScores*
        - Input: None
        - Output: Round Name; Season; Year; Contestant Name; Current Score
        - Parent Dependency: JudgeMenu
        - Child Dependency: None
    - *ContestantMenu*
        - Input: None

- Output: None
- Parent Dependency: Login
- Child Dependency: ContestantCalendarOfEvaluation, ContestantReportOfRounds, ContestantReportOfScores

■ *ContestantCalendarOfEvaluation*
- Input: None
- Output: Round Name; Evaluation Type; Deadline
- Parent Dependency: ContestantMenu
- Child Dependency: None

■ *ContestantReportOfRounds*
- Input: None
- Output: Round Name; Season; Year; Time; Location
- Parent Dependency: ContestantMenu
- Child Dependency: None

■ ContestantReportOfScores
- Input: None
- Output: Round Name; Evaluation Type; Percent Weight; Current Score
- Parent Dependency: ContestantMenu
- Child Dependency: None

➢ Incremental Testing (Top-down)

- Top down incremental testing
  - Top-down incremental testing is a testing method which begins by testing the top level module, and progressively adds in lower level modules
  - Lower level modules are normally simulated by stubs which mimic functionality of lower level modules. As you add lower level code, you will replace stubs with the actual components.
- We use the top down form because
  - Driver do not have to be written when top down testing is used.
  - It provides early working module of the program and so design defects can be found and corrected early.

Test #	Testing Component	Test Case	Severity	Solution
1	MainFrame	**Input:** Start program **Output:** A frame should appear	1	

2	MainFrame *LoginScreen*	**Input:** User ID; Password; Select “Judge” button; Click “Login” button. **Output:** If credential is correct, switch to JudgeMenuScreen; otherwise, popup box saying “ERROR: invalid credential” or “ERROR: invalid input”.	3	
3	MainFrame *LoginScreen*	**Input:** User ID; Password; Select “Contestant” button; Click “Login” button. **Output:** If credential is correct, switch to JudgeMenuScreen; otherwise, popup box saying “ERROR: invalid credential” or “ERROR: invalid input”.	3	
4	MainFrame LoginScreen *JudgeMenuScreen*	**Input:** Repeat input from Test #2; Click on menu buttons. **Output:** Corresponding screen to each button should appear.	2	
5	MainFrame LoginScreen *ContestantMenuScreen*	**Input:** Repeat input from Test #3; Click on menu buttons. **Output:** Corresponding screen to each button should appear.	2	
6	MainFrame LoginScreen JudgeMenuScreen *JudgeCreateRoundScreen*	**Input:** Repeat input from Test #4; Name; Season; Year; Time; Location; Click “Submit” button. **Output:** If record is successfully added to database, popup box saying “SUCCESS:	1	Correct corresponding query statement

		record added”; otherwise, popup box saying “ERROR: failed to add record”.		
7	MainFrame LoginScreen JudgeMenuScreen JudgeCreateRoundScreen *JudgeModifyRoundScreen*	**Input:** Repeat input from Test #6; Select the round that just created;Name; Period; Year; Meets At; TestRoom; Click “Submit” button. **Output:** If record is successfully modified in database, popup box saying “SUCCESS: record modified”; otherwise, popup box saying “ERROR: failed to modify record”.	2	Correct corresponding query statement
8	MainFrame LoginScreen JudgeMenuScreen JudgeCreateRoundScreen JudgeModifyRoundScreen *JudgeAssignContestantToRound*	**Input:** Repeat input from Test #7; Select Contestant ID and Round name; Click “Submit” button; **Output:** If record is successfully added to database, popup box saying “SUCCESS: record added”; otherwise, popup box saying “ERROR: failed to add record”.	1	Correct corresponding query statement
9	MainFrame LoginScreen JudgeMenuScreen JudgeCreateRoundScreen JudgeModifyRoundScreen JudgeAssignContestantToRound *JudgeCreateAnEvaluation* *JudgeModifyAnEvaluation*	**Input:** Repeat input from Test #8; Click “Create An Evaluation”: Select “Round”; Evaluation Type; Percent Weight; Deadline; Click “Submit”; Click “Modify An Evaluation”; Select “Round” and “Evaluation Type”; Percent Weight; Deadline; Click “Submit”. **Output:** If record is successfully modified in database, popup box saying “SUCCESS: record modified”; otherwise, popup box saying “ERROR: failed to modify record”.	1	Correct corresponding query statement
10	MainFrame LoginScreen JudgeMenuScreen JudgeCreateRoundScreen	**Input:** Repeat input from Test #9; Click “Enter Score”: Select Round, Evaluation, Contestant;	1	Correct corresponding query statement

	JudgeModifyRoundScreen JudgeAssignContestantToRound JudgeCreateAnEvaluation JudgeModifyAnEvaluation *JudgeEnterScore*	Score; Click “Submit”. **Output:** If record is successfully added to database, popup box saying “SUCCESS: record added”; otherwise, popup box saying “ERROR: failed to add record”.		
11	MainFrame Login Screen JudgeMenuScreen JudgeCreateRoundScreen JudgeModifyRoundScreen JudgeAssignContestantToRound JudgeCreateAnEvaluation JudgeModifyAnEvaluation JudgeEnterScore *JudgeReportOfRounds* *JudgeReportOfContestantsAndScores*	**Input:** Repeat input from Test #10 Click “Report of Round”; Click “Report of Contests and scores” **Output:** Report tables showing query results from database.	1	Correct corresponding query statement
12	MainFrame LoginScreen ContestantMenuScreen *ContestantCalendarOfEvaluation*	**Input:** Repeat input from Test #5; Click “Calendar of Evaluation”. **Output:** Display the contestant’s own evaluations calendar as a table.	1	Correct corresponding query statement
13	MainFrame LoginScreen ContestantMenuScreen ContestantCalendarOfEvaluation *ContestantReportOfRounds*	**Input:** Repeat input from Test #12; Click “Report of Rounds”. **Output:** Display info of all rounds that the contestant participates in.	1	Correct corresponding query statement
14	MainFrame LoginScreen ContestantMenuScreen ContestantCalendarOfEvaluation ContestantReportOfRounds *ContestantReportOfScores*	**Input:** Repeat input from Test #13; Click “Report of Scores”. **Output:** Display the contestant’s score of all rounds participated in.	1	Correct corresponding query statement

15	MainFrame LoginScreen JudgeMenuScreen JudgeCreateRoundScreen JudgeModifyRoundScreen JudgeAssignContestantToRound JudgeCreateAnEvaluation JudgeModifyAnEvaluation JudgeEnterScore JudgeReportOfRounds JudgeReportOfContestantsAndScores ContestantMenuScreen ContestantCalendarOfEvaluation ContestantReportOfRounds ContestantReportOfScores	**Input:** Start program; Log in as Judge; Create a “Round 1”; Assign “Contestant X” to “Round 1”; Modify “Round 1”’s attributes; Create an “Evaluation 1” for “Contestan X” in “Round 1”; Enter Score for “Evaluation 1”; Click “Report of Rounds” to check correctness of “Round 1”; Click “Report of Contestant and Scores” to check correctness of “Contestant 1”’s score in “Round 1”; Logout; Log in as “Contestant X”; Click “Calendar of Evaluations” to check correctness of “Evaluation 1” info; Click “Report of Rounds” to check correctness of “Round 1” info; Click “Report of Scores” to check correctness of score in “Round 1”; **Output:** All the data should be consistent.	1	Correct correspon ding query statement

➢ Regression Testing

- Create A Round

  “ModifyARound”, “ssignContestantToARound”, “CreateAnEvaluation”, “ModifyAnEvaluation”, and “Report of Rounds” are logically affected by changes of “CreateARound”. So if defects revealed in “CreateARound” are corrected, regression test cases should be run on above components to verify that they also work correctly.

- Create An Evaluation

  “ModifyAnEvaluation”, ”EnterAScore”, and ReportsOfContestantsAndScores” are logically affected by changes of “CreateAnEvaluation”. So if defects revealed in “CreateAnEvaluation” are corrected, regression test cases should be run on above components to verify that they also work correctly.

- Assign Contestant to Round

"AssignContestanttoRound", "CreateARound" are logically affected by changes of "CreateARound". So if defects revealed in "CreateARound" are corrected, regression test cases should be run on above components to verify that they also work correctly.

- Enter Score

"JudgeReportOfContestantsAndScores", "ContestantReportOfScores" are logically affected by changes of "EnterScore". So if defects revealed in "EnterScore" are corrected, regression test cases should be run on above components to verify that they also work correctly.