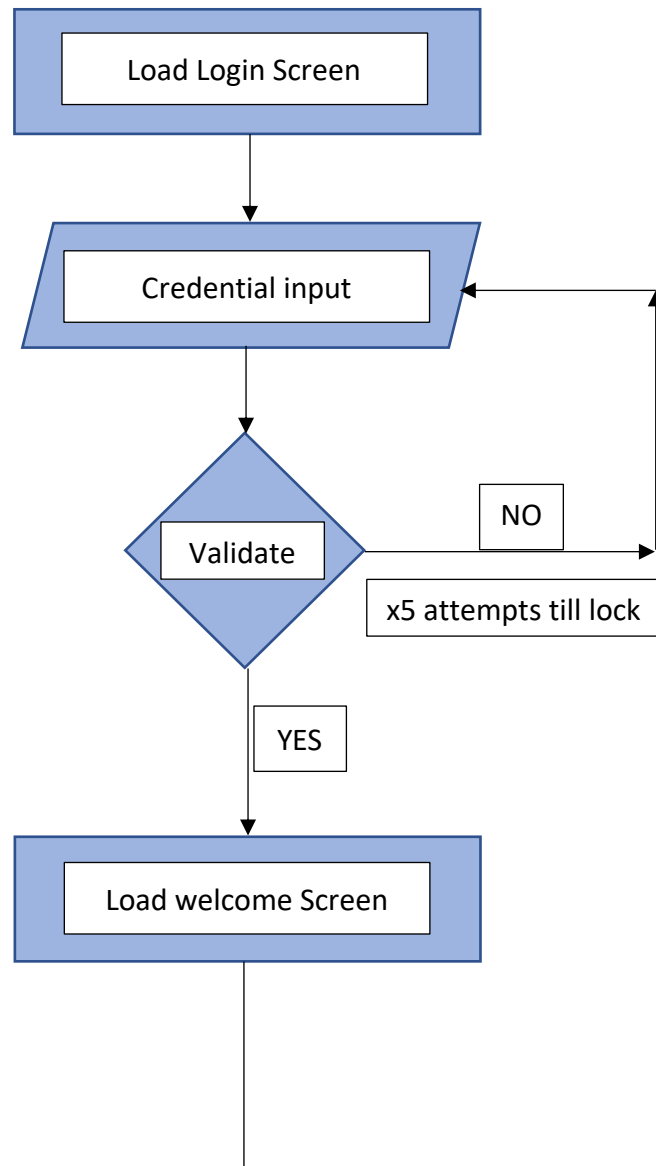
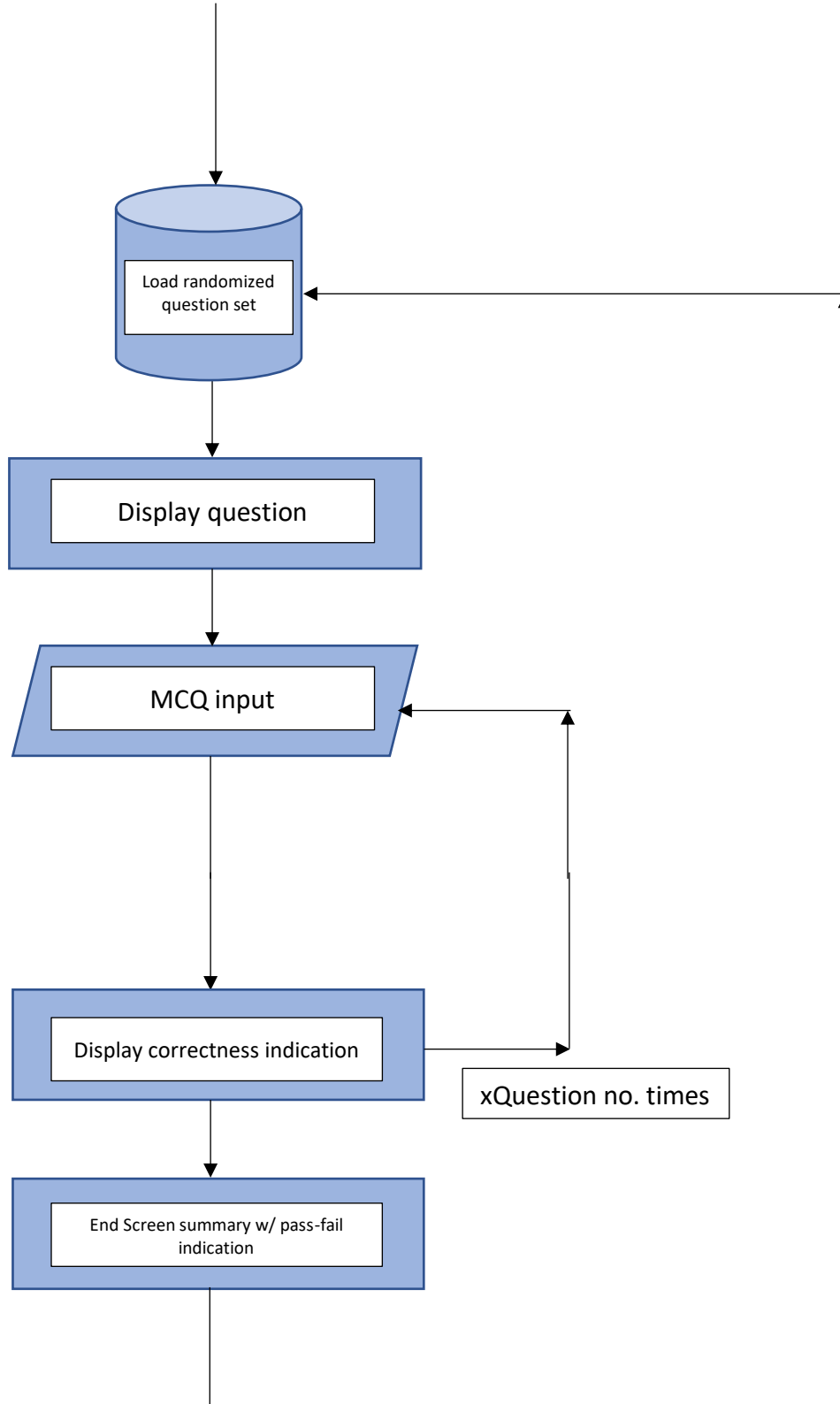


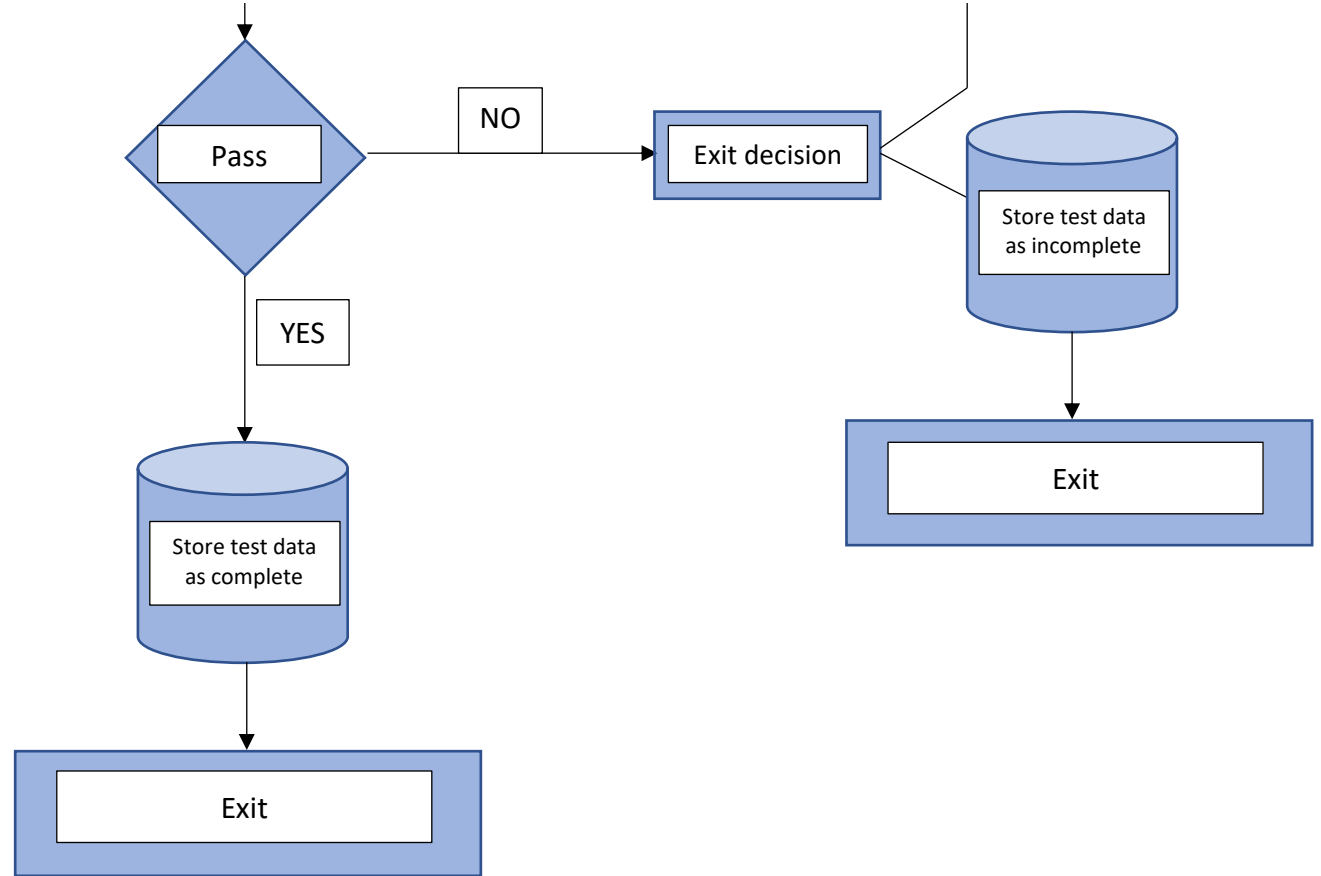
Criterion B: Design

UX Flow chart and Design

Flowchart for Student

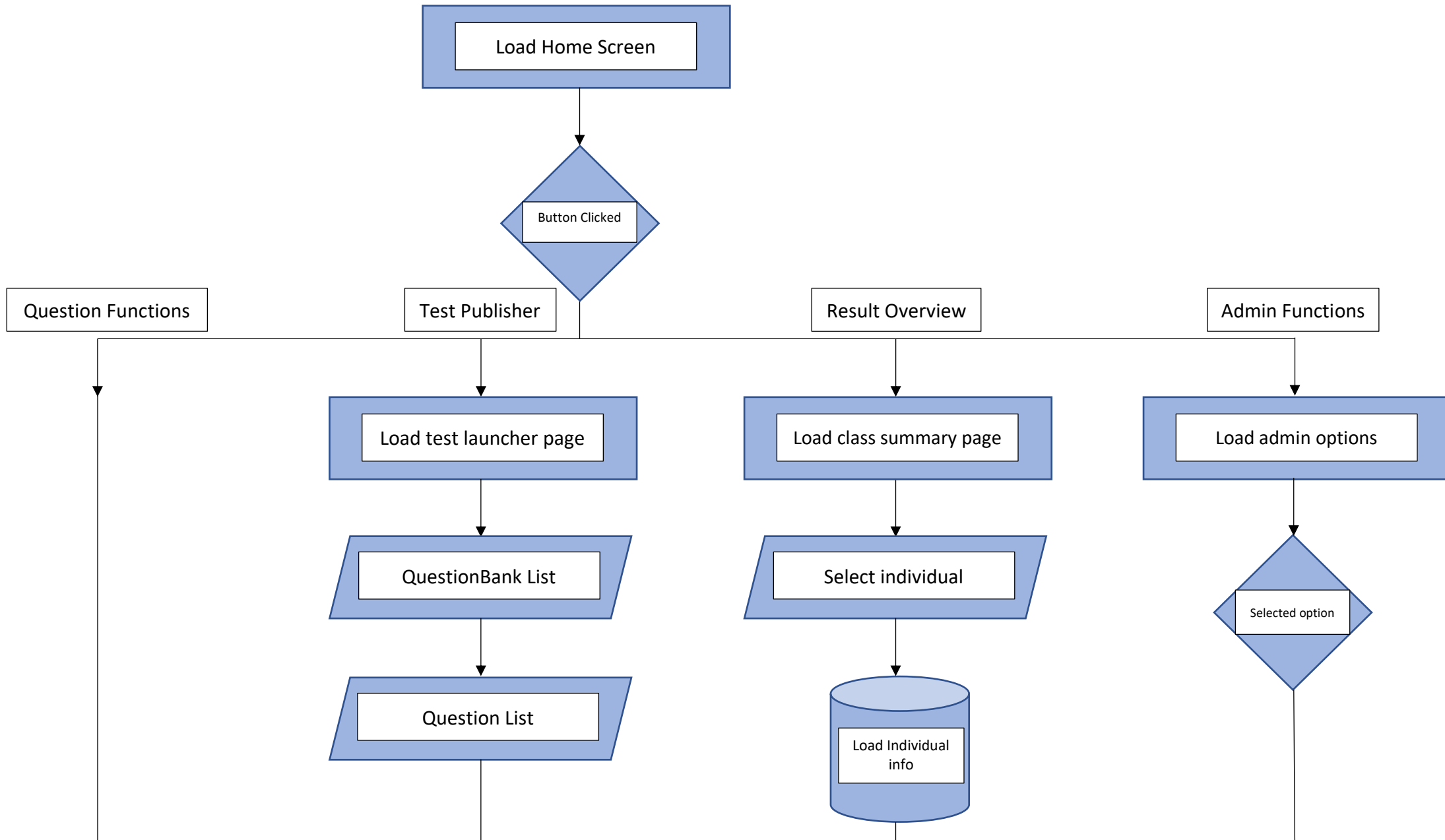


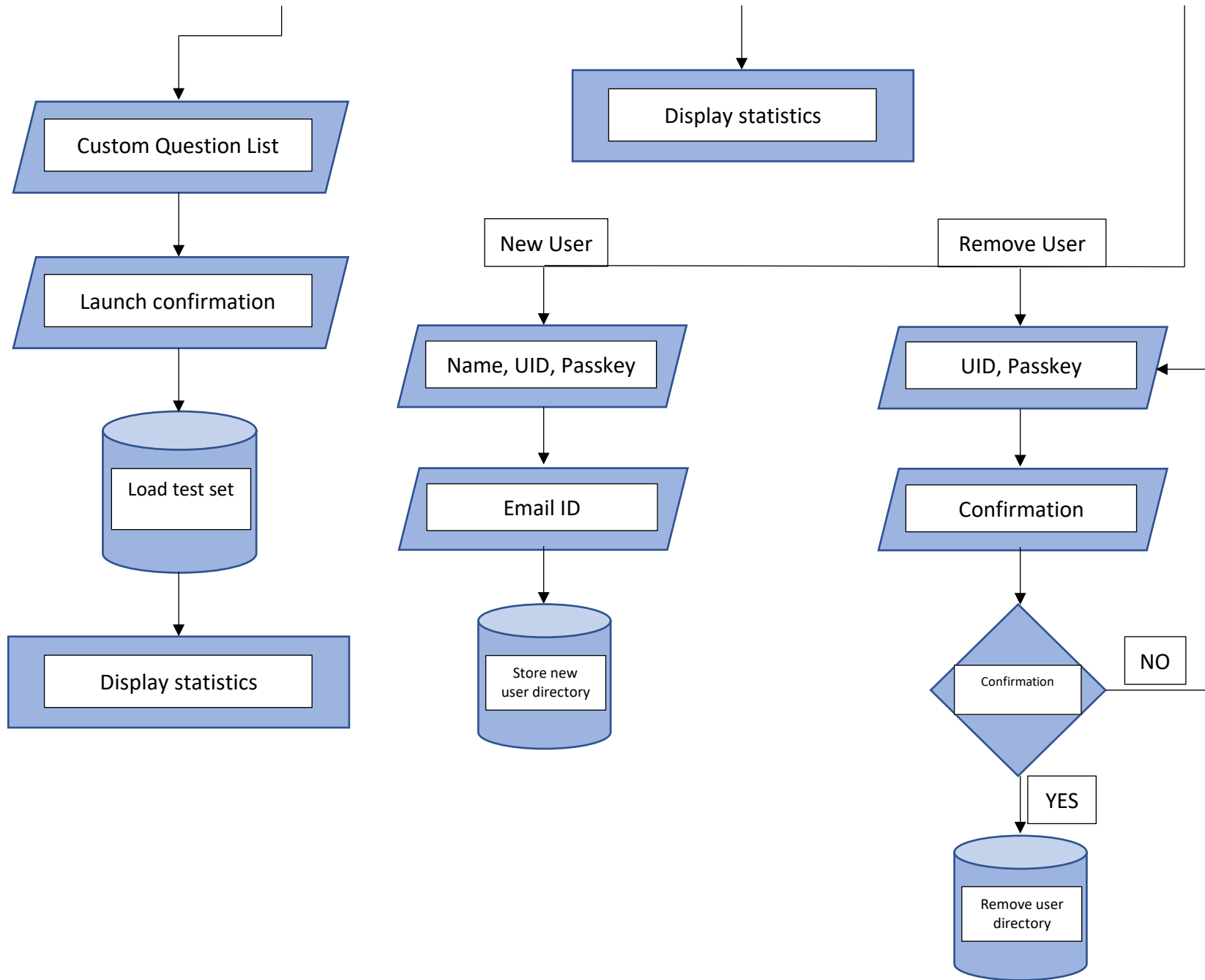


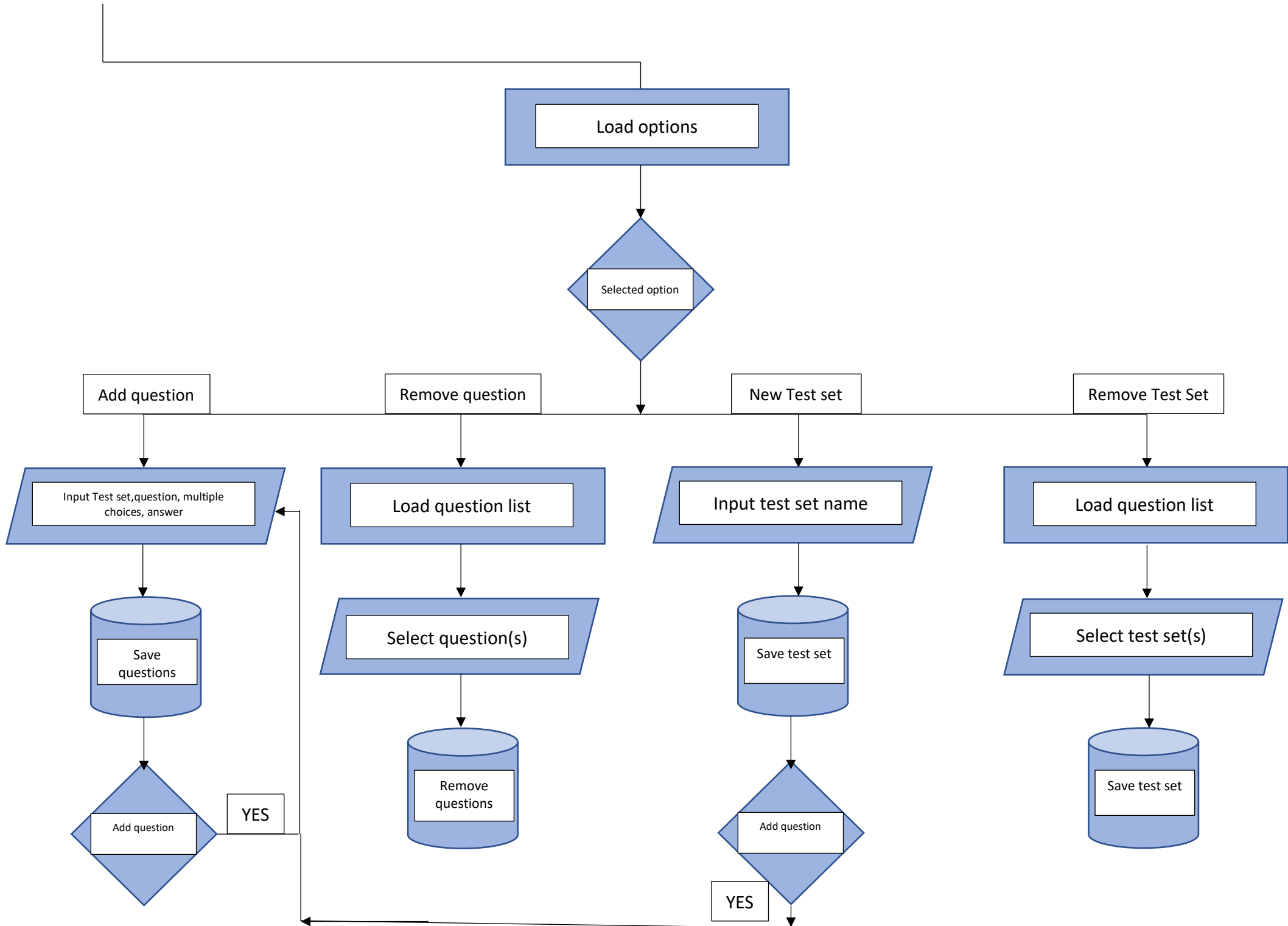


UX Flow chart and Design

Flowchart for Supervisor/Admin







Criterion B: Design Overview

Hardware Requirements:

- A desktop, laptop or remote server

Software Requirements:

- MacOS 10.X, Windows 7 (or higher) or GNU/Linux(Arch Linux)
- Python 3.6.5 or higher

UX Proposal:

Login Screen:

- Username field: Username Entry box field for users (Employees and Supervisors alike) to enter username credentials.
- Password field: Password Entry box field for users Employees and Supervisors alike) to enter password credentials.

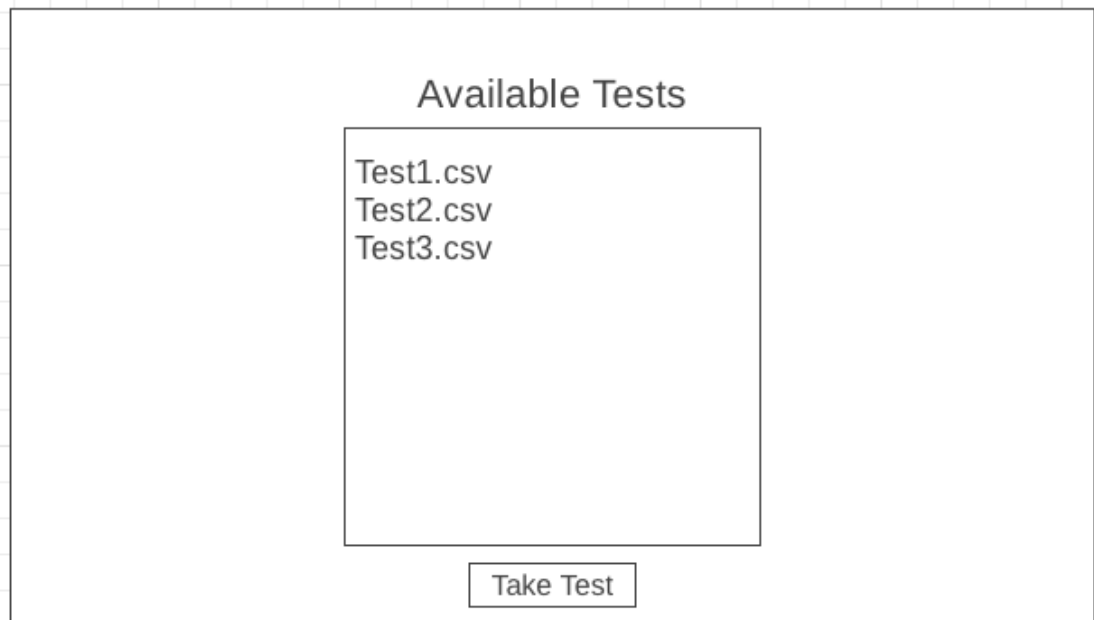


The image shows a sample login screen for ECI Telecom. It features a large blue logo at the top center. Below the logo, there are two input fields: one for the username labeled 'User:' and one for the password labeled 'Password:'. Both fields are empty. Below the password field, there is a blue 'Login' button.

Image 1: Sample Login Screen

Available Tests (Employee only):

- Test List Box: Test list box displays all the available published tests ready for the employee to take
- Button: Button to submit choice of test response

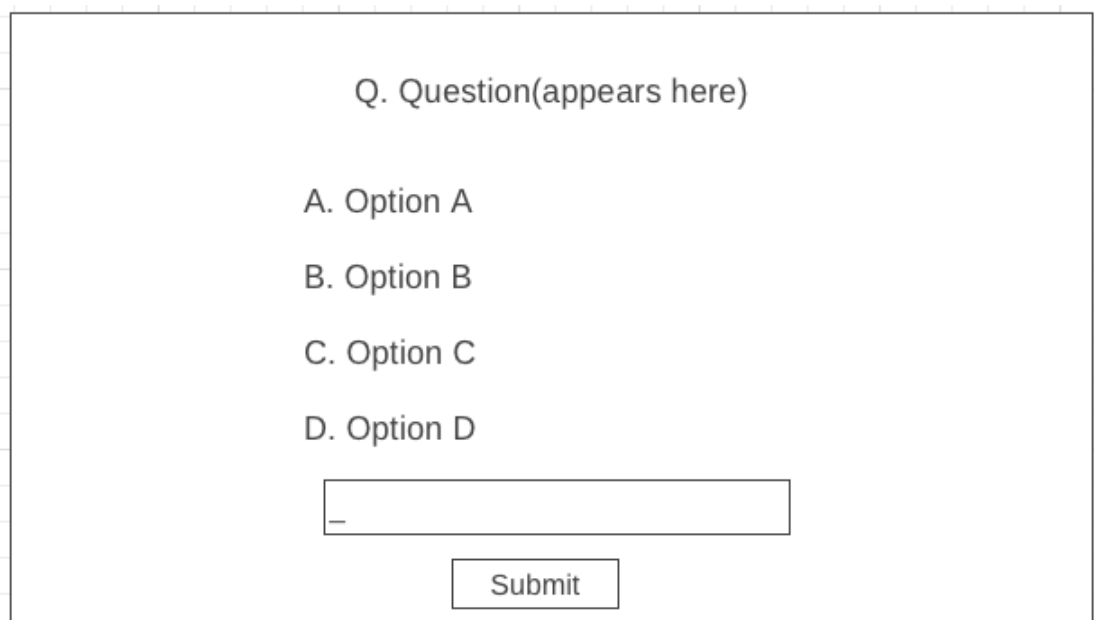


A screenshot of a web application interface titled "Available Tests". It features a central rectangular box containing a list of three test files: "Test1.csv", "Test2.csv", and "Test3.csv". Below this box is a button labeled "Take Test". The entire interface is set against a light gray background with a subtle grid pattern.

Image 2: Sample Test Choice Screen

Test Take Screen (Employee only):

- Labels: Text labels that display question, multiple choice options to present test question.
- Entry Box: Entry Box to provide response to answer



A screenshot of a web application interface for taking a test. It displays a question labeled "Q. Question(appears here)". Below the question are four multiple-choice options: "A. Option A", "B. Option B", "C. Option C", and "D. Option D". At the bottom of the screen is a text entry box with a cursor and a button labeled "Submit". The interface has a light gray background with a subtle grid pattern.

Image 3: Question Response and Test Taker Screen

Software Functions Screen (Supervisor Only):

The image shows a software interface titled "ECI Telecom". Below the title, there are four rectangular buttons stacked vertically. The buttons are labeled "Admin Functions", "Test Publish", "Question Functions", and "Employee Stats".

ECI Telecom

Admin Functions

Test Publish

Question Functions

Employee Stats

Image 4: Software Functions Screen

- Admin Functions: Admin Functions button to redirect user to screen containing various admin-oriented functions such as create new user/remove user.

The image shows a software interface titled "Admin Functions". Below the title, there are two sections. The first section is labeled "Add User:" and contains four input fields followed by a "Submit" button. The second section is labeled "Remove User:" and contains two input fields followed by a "Submit" button.

Admin Functions

Add User:

Submit

Remove User:

Submit

Image 5: Admin Functions Screen

- Test Publish: Test publish button to redirect users to screen containing various test publishing related functions, such as choice of test bank and various questions to publish to test-taking employees

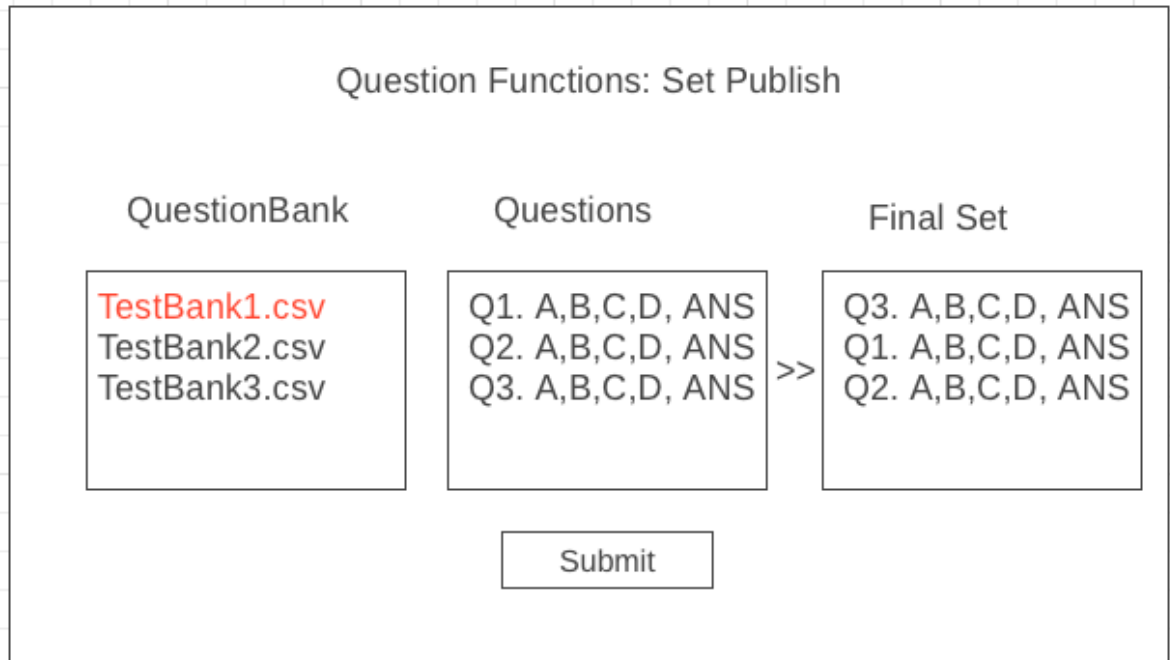


Image 6: Question Functions Screen – Test Publish

- Question Functions: Question functions button to redirect user to screen containing question managing functions, such as creation/removal of new test banks and addition/removal of questions from said test banks.

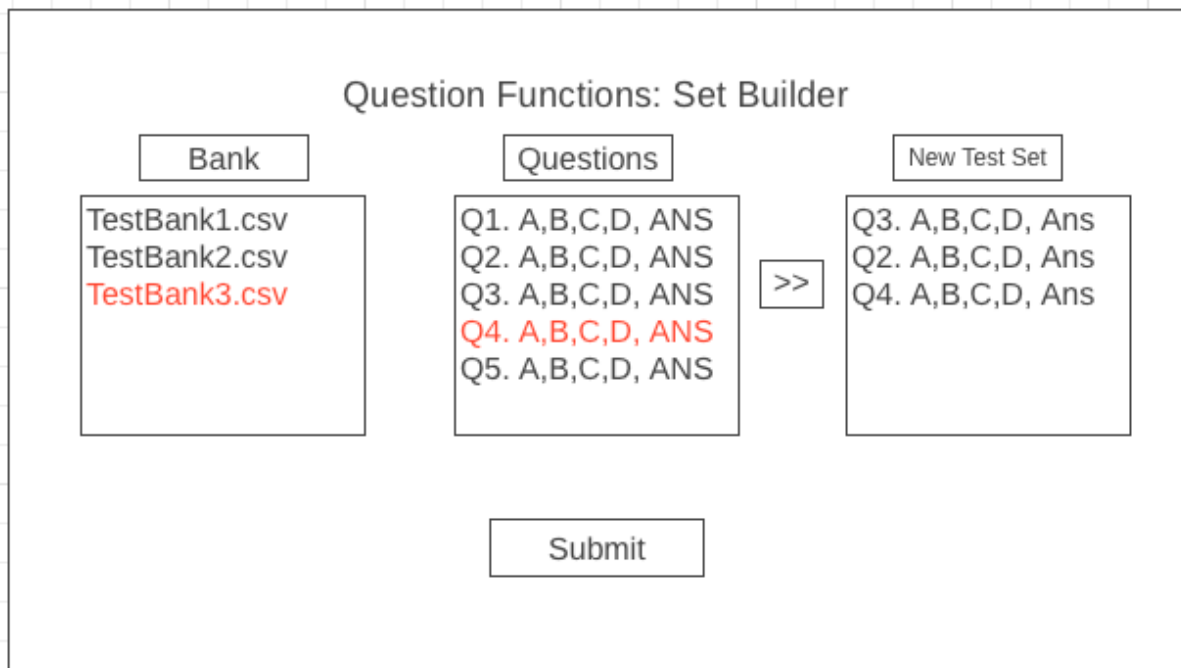


Image 7: Question Functions Screen – Test Question Remover (1/3)

Question Functions: Add Bank new Question

Question:

Option A:

Option B:

Option C:

Option D:

Image 8: Question Functions Screen – Test New Question Add (2/3)

Question Functions: Add Bank

TestBank

Add or Remove?

Image 9: Question Functions Screen – Test New Bank Add (3/3)

- Employee Stats: Employee Stats button to redirect user to screen containing overview of various test-takers and an overview of their performance by their choices.

Employee Stats

Employees

Sam
Elliot
Angela
Darlene

Tests Taken

set1.csv
set4.csv
set3.csv

Submit

Image 10: Employee Overview Screen – Test Employee List View + Tests (1/2)

Employee Stats

Ref no., Question, Options, Response

Candidate Response

TestBank Ans

1,Ques, A,B,C,D, Resp.
5,Ques, A,B,C,D, Resp.
3,Ques, A,B,C,D, Resp.
7,Ques, A,B,C,D, Resp.

1,Ques, A,B,C,D, Ans.
2,Ques, A,B,C,D, Ans.
3,Ques, A,B,C,D, Ans.
4,Ques, A,B,C,D, Ans.

Image 11: Question Overview Screen – Test Employee Status Overview (2/2)

Test Functionality	Procedure
1. Login function	<ol style="list-style-type: none"> 1. Enter an existing user's credentials 2. Enter credentials with spelling error 3. Check if appropriate error messages, files and updates to existing files are made 4. Enter invalid credential (invalid character and non-user cred) to test login.
2. Profile Creation	<ol style="list-style-type: none"> 1. Enter credential to create new user 2. Enter Credentials to remove existing user 3. Check if appropriate error messages, files and updates to existing files are made 4. Enter invalid user credentials in remove user and duplicate credentials in create user to test error messages.
3. Available Test View	<ol style="list-style-type: none"> 1. View shown available test 2. Check if all available test are being displayed
4. Take Test Function	<ol style="list-style-type: none"> 1. Answer Questions as indicated 2. Reach End of Test 3. Check if responses correctly recorded in Employee Overview
5. Software Function Testing	<ol style="list-style-type: none"> 1. Test if various buttons redirect to correct pages in the software functions page by checking each button individually for redirect.
6. Add/Remove Question Testing	<ol style="list-style-type: none"> 1. Fill in the various entry boxes (question, options, answer) appropriately, after selecting the desired test bank to add question. Select various questions and use remove button to remove questions from Listview Box. 2. Check to see if TestBank was updated with new Questions by reopening window. 3. Check to see if TestBank was updated with old Questions by reopening window..

7. Add/Remove TestBank	<ol style="list-style-type: none"> 1. Fill name of TestBank and choice of Add/Remove TestBank and abnormal data. 2. Check filesystem for appropriate removal/addition of TestBank and appropriate error message otherwise
8. Employee Status	<ol style="list-style-type: none"> 1. Select Various Employees to see whether various taken tests reflect actual tests taken by checking appropriate Filesystem files. 2. Choose a test and submit to check whether given test (responses + options+ ref no.) and answers(questions+options+answers) correctly appear, and check Filesystem if correct files are appearing

EDIT:

After demonstrating the product to the client and acquiring relevant feedback, I have made some updates to the login screen of the program. The program now accepts an additional field for a special 'Master Key' that when typed in the hidden field, allows Admin/Supervisor to directly access the Software functions page.

ECI TELECOM Login Page

Username:

Password:

Supervisor Login:

Image 12: Login Screen (Updated with Supervisor Login box)

UML Diagrams:

TestGUI_Interface_loginScr

+ __init__(self, master): main
+ MasterBox(self)
+ submit(self):

TestGUI_Interface_SuperInterface_PublishTest

+ __init__(self, master): main
+ TBankSelect(self)
+ TBankQuestionSelect(self)
+ Done(self)

TestGUI_Interface_SuperInterface_AdF

+ __init__(self, master): main
submit(self)
submit2(self)

TestGUI_Interface_SuperInterface

+ __init__(self, master): main
+ nextWindow(self)
+addRemove_Submit(self)
+Publish(self)
+Employee(self)

TestGUI_Interface_SuperInterface_QuestionManager

– SessionBankNameVariable: String
+ submitTBankSelection_Function(self)
+ removeQuestion_Function(self)
+ submitAddQuestion(self)
+ polyTestSet(self)
+ __init__(self, master): main

TestGUI_Interface_SubInterface_TestTake

+ __init__(self, master): main
+ Submit(self)