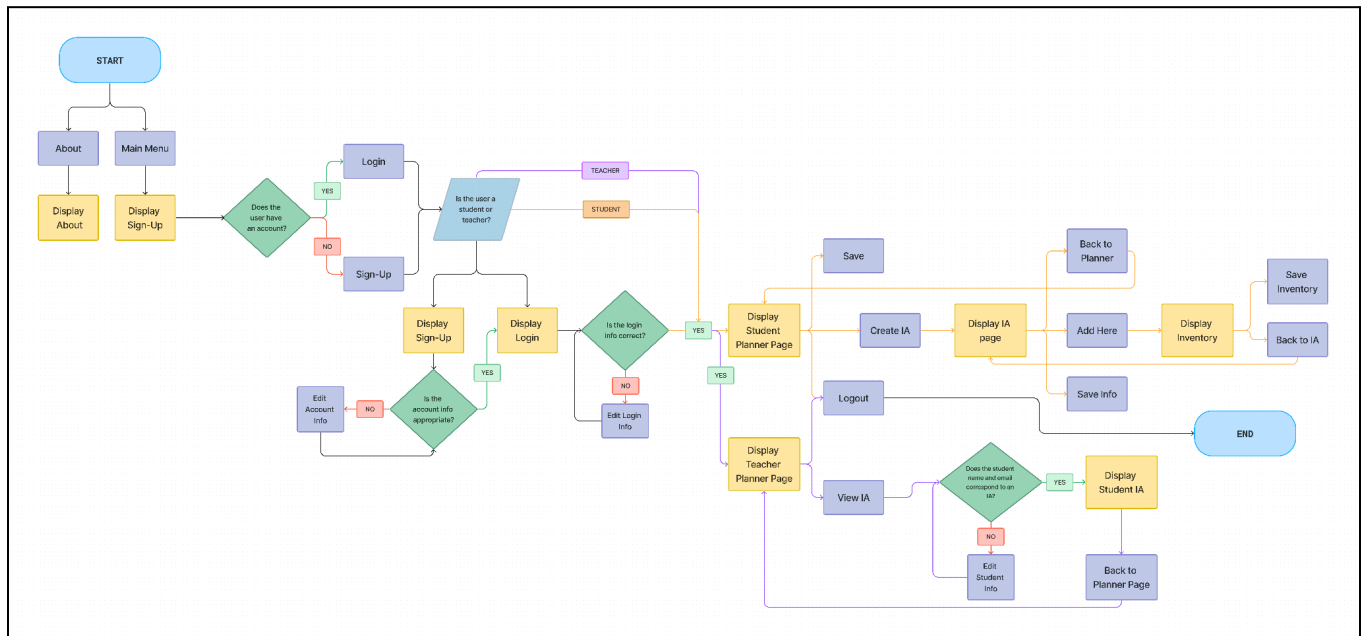


Criterion B: Design

Test Plan

Test Type	Nature of Test	Example
When program is entered, user should be able to click on the about or main menu page	Check that both buttons lead to the appropriate frames of the application	<ul style="list-style-type: none"> - User clicks on “about”, info about the app appears - User clicks “main menu”, enters application menu
Check that user can sign-up and login as a student or teacher	User enters a username and password for either profile	<ul style="list-style-type: none"> - If user is a student, directed to a planning page - If user is a teacher, directed to the teacher profile for viewing student work
Check that sign-up and login validate user’s account info and saves it appropriately	User can enter any username or password and the application verifies the account	<ul style="list-style-type: none"> - User enters password < 8 characters, password error message - User enters wrong name, account error message
If the user is logged in as a student, they should be able to enter their personal and IA information (saves to a personal text file)	Check if the user fills the fields and then write the correct information into the text file	<ul style="list-style-type: none"> - If user enters “The effects of HCL on a potato”, text file should return: <i>Topic: The effects of HCL on a potato</i>
Check if the student user can save the items they choose from the materials list	Test the addition of the materials list by clicking the save inventory button	<ul style="list-style-type: none"> - If user clicks save inventory, the text file should return: <i>Materials:</i> <ul style="list-style-type: none"> - <i>Hydrochloric acid</i> - <i>Beakers</i>
If the user is logged in as a teacher, they can access the student’s IA plan by entering the student’s username and student number	Check if both fields are filled correctly, and if true, return the appropriate IA info of the particular student	<ul style="list-style-type: none"> - If user enters “Alex” & 3472019234, frame should display IA info of Alex
If the user clicks the logout button on their respective frames, they should be logged out of the application	Check the display of the logout button on click	<ul style="list-style-type: none"> - If user clicks “logout”, frame displays “you’ve successfully logged out. see you later!”

Program Flowchart



Graphical Visualization

Welcome Page - DON'T BOMB IT

- > Click About
 - > Learn more about the application
 - > BACK
- > Click Main Menu
 - > Sign-Up
 - > Is the User a STUDENT OR TEACHER?
 - > Enter Account Info (username & password w/ min 8 characters)
 - > Validate and confirm sign-up -> Head to Login
 - > Login
 - > Validate and confirm login info

AFTER LOGIN IS CONFIRMED

- > If user is a STUDENT:
 - > Hello, user's name
 - > header: your planner
 - > STUDENT NUMBER:
 - > STUDENT EMAIL:
 - > STUDENT LEVEL:
 - > CREATE IA
 - > header: the IA
 - > TOPIC:
 - > RESEARCH QUESTION:
 - > HYPOTHESIS:
 - > MATERIALS: ADD HERE
 - > open new panel (select materials)
 - > click SAVE INVENTORY
 - > VARIABLES:
 - > SAVE INFO
 - > SAVE
 - > LOGOUT

AFTER IS LOGIN IS CONFIRMED

-> If user is a TEACHER:

-> Hello, user's name

-> header: your planner

-> Enter the student's username and email to view their IA

-> STUDENT NAME:

-> STUDENT EMAIL:

-> VIEW IA - leads to selected student's IA

-> BACK TO PLANNER (heads back to teacher planner)

-> LOGOUT

Design Progress

The image shows three panels for a web application. The top-left panel is a welcome message with the text "don't bomb it" and "the chem IA planner you didn't know you needed", with links for "about" and "main menu". The bottom-left panel is a sign-up form titled "sign-up" with the instruction "select the appropriate profile" and two icons: a graduation cap and a person at a computer. The right panel is a login form titled "login" with the instruction "enter the appropriate info", fields for "username" and "password", and an "enter" button.

Initial design for welcome, sign-up, & login panels

The image shows three panels for a web application. The top-left panel is a student planner form with fields for "student #:", "student email:", "select level:" (with radio buttons for "SL" and "HL"), and "create your IA here:". The top-right panel is a materials list with checkboxes for "Forceps/Tongs", "Graduated cylinder", "Thermometer", "100mL/500mL beaker", "Erlenmeyer/Volumetric flask", and "Funnel". The bottom panel is the IA panel titled "the IA" with fields for "topic:" and "research question:".

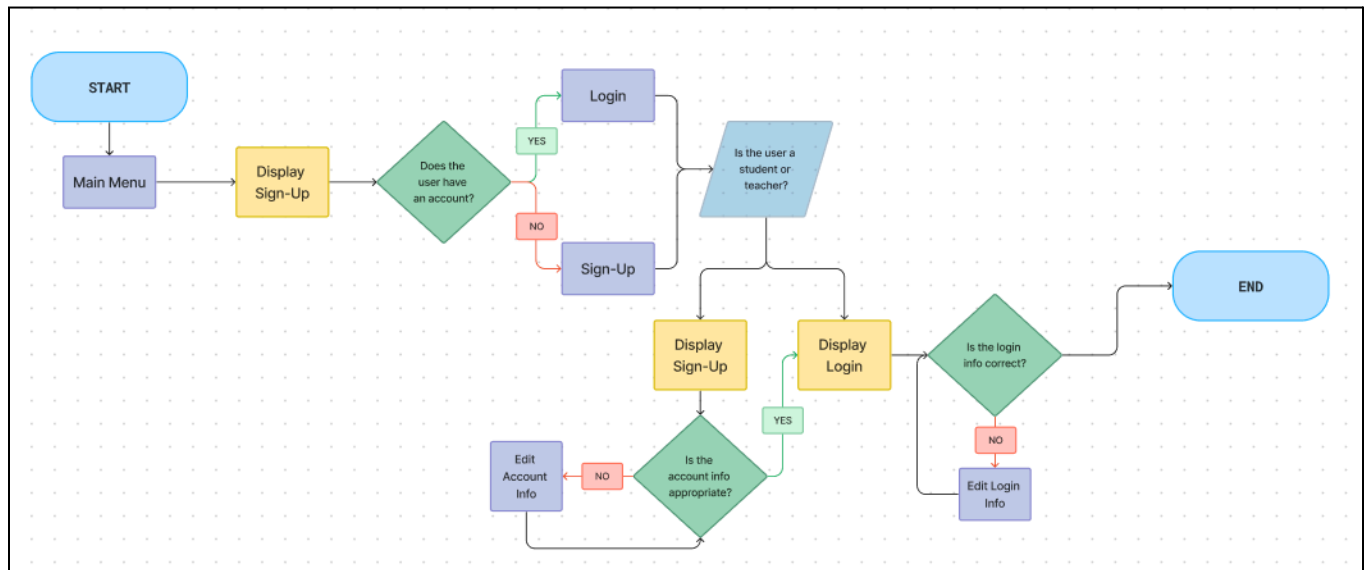
Initial panel design ideas for student planner, IA panel, & materials list

The image shows two panels for a web application. The top panel is a teacher planner form titled "enter student username and email to view" with fields for "student name:", "AND", and "student number:", and a "view IA" button. The bottom panel is an IA review panel showing student information and materials. The student information includes: Student: sri, Student #: 335097101, Email: sriakshini.akhilam@student.tdsb.on.ca, Level: SL. The materials list includes: 100mL/500mL beaker, Erlenmeyer/Volumetric flask, Pipettes/Droppers, and Safety Gear.

Initial designs for teacher planner & IA review panel

Algorithm Explanations

Sign-Up/Login



Inputs: String - USERFIELD, PWDFIELD

Variables: JLabel - INFOSTATUS, HashMap - INFO<String, GENERALUSER>

// Sign-Up Constructor

```
if USERFIELD.LENGTH() < 1 OR PWDFIELD.LENGTH() < 8 then
    INFOSTATUS = "seems like your info is incorrect, try again"
else
    INFOSTATUS.SETTEXT("")
    INFO.PUT(USERFIELD, NEW STUDENTUSER(USERFIELD, PWDFIELD))
end if
```

Sign-Up

```
// Login Constructor
try
    BufferedReader READ = new BufferedReader(new FileReader(ACCNTFILE))
    loop while READLINE = READ.READLINE() != NULL
        ACCOUNTINFO = READLINE.SPLIT("") // string methods doesn't exist in pseudocode
        USERNAME = ACCOUNTINFO[0]
        PASSWORD = ACCOUNTINFO[1]
        USERTYPE = ACCOUNTINFO[2]

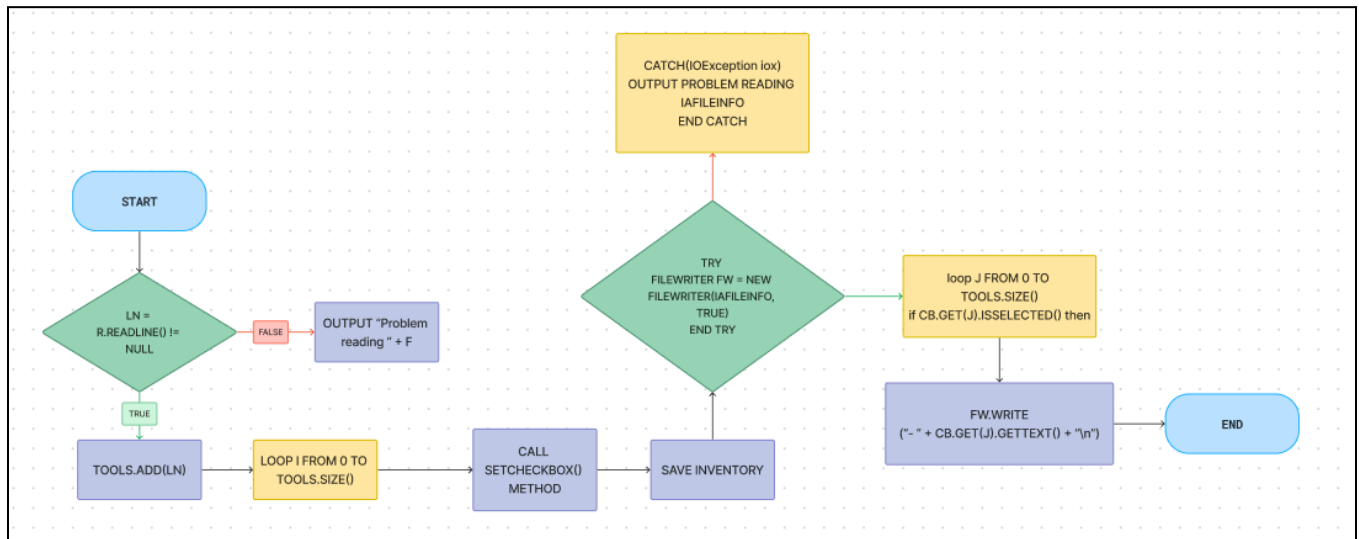
        if USERTYPE.equals("student") && INFO.GET(USERNAME) == NULL then // string methods doesn't exist in pseudocode
            INFO.PUT(USERNAME, NEW STUDENTUSER(USERNAME, PASSWORD))
        else if INFO.GET(USERNAME) == NULL then
            INFO.PUT(USERNAME, NEW TEACHERUSER(USERNAME, PASSWORD))
        end if
    end loop

    READ.CLOSE()
end try

catch(IOException IOX)
    OUTPUT "Problem reading " + ACCNTFILE
end catch
```

Login

Inventory List



```
// Material List Panel
Variables: JCheckBox - MATERIALS, String - F, LINE, ArrayList<String> - TOOLS, ArrayList<JCheckBox> - CB, JPanel - JP

try
    BufferedReader R = new BufferedReader(new FileReader(F))
    loop while LINE = R.READLINE() != NULL
        TOOLS.ADD(LINE)
    end loop

    R.CLOSE()
end try

catch(IOException IOX)
    OUTPUT "Problem reading " + F
end catch

loop I from 0 to TOOLS.SIZE()
    MATERIALS = NEW JCHECKBOX()
    MATERIALS = GUICOMP.SETCHECKBOX(MATERIALS, TOOLS.GET(I), 0, 0, 0, 0)

    JP.ADD(MATERIALS)
    CB.ADD(MATERIALS)
end loop
```

```
// Saving Inventory from Material List
Variables: JButton - SAVEINVBTN, String - IAFILEINFO, FileWriter - FW, ArrayList<String> - TOOLS, ArrayList<JCheckBox> - CB

SAVEINVBTN == PRESSED

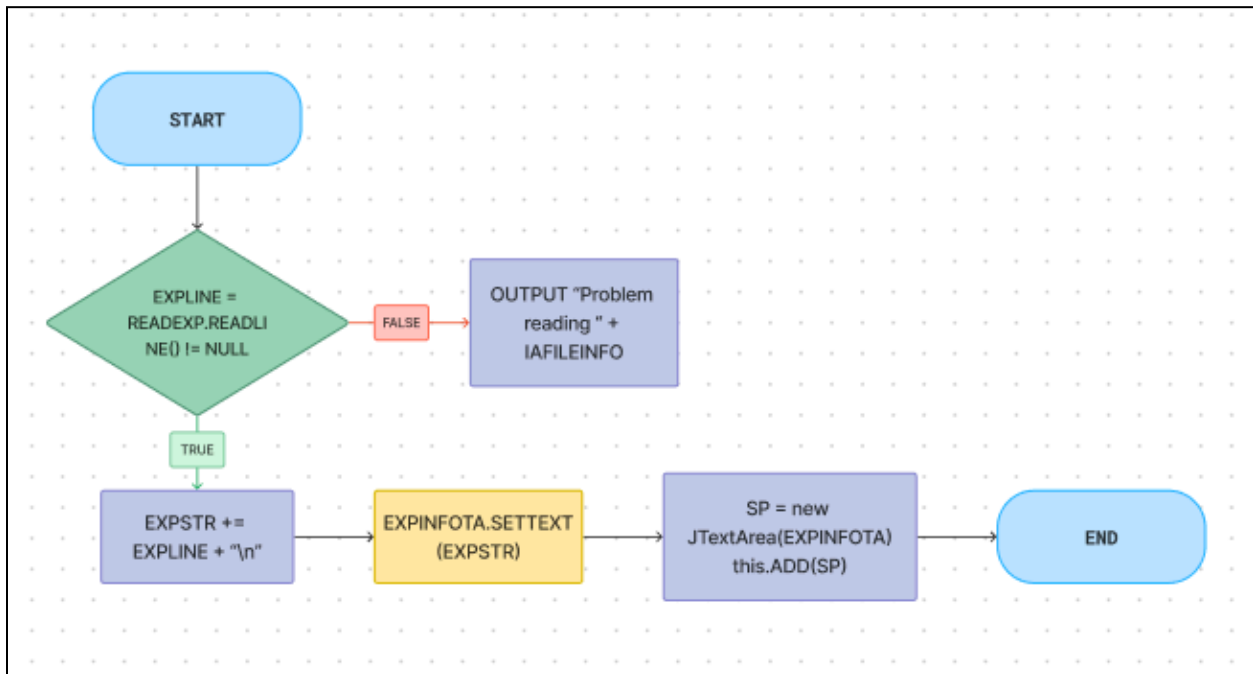
try
    FileWriter FW = new FileWriter(IAFILEINFO, true)
    FW.WRITE("Materials: " + "\n")

    loop J from 0 to TOOLS.SIZE()
        if CB.GET(J).ISSELECTED() then
            FW.WRITE("- " + CB.GET(J).GETTEXT() + "\n")
        end loop

    FW.CLOSE()
end try

catch(IOException IOX)
    OUTPUT "Problem reading " + IAFILEINFO
end catch
```

Review IA Panel



```
// Review IA Panel
Variables: BufferedReader - READEXP, String - IAFILEINFO, EXPLINE, EXPSTR, JTextArea - EXPINFOTA, JScrollPane - SP

EXPINFOTA = NEW JTextArea(EXPSTR)
SP = NEW JScrollPane(EXPINFOTA, JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED, JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED)

try
    BufferedReader READEXP = new BufferedReader(new FileReader(IAFILEINFO))
    loop while EXPLINE = READEXP.READLINE() != NULL
        EXPSTR += EXPLINE + "\n"
    end loop
    READEXP.CLOSE()
end try

catch(IOException IOX)
    OUTPUT "Problem reading " + IAFILEINFO
end catch

EXPINFOTA.SETTEXT(EXPSTR)
this.ADD(SP)
```