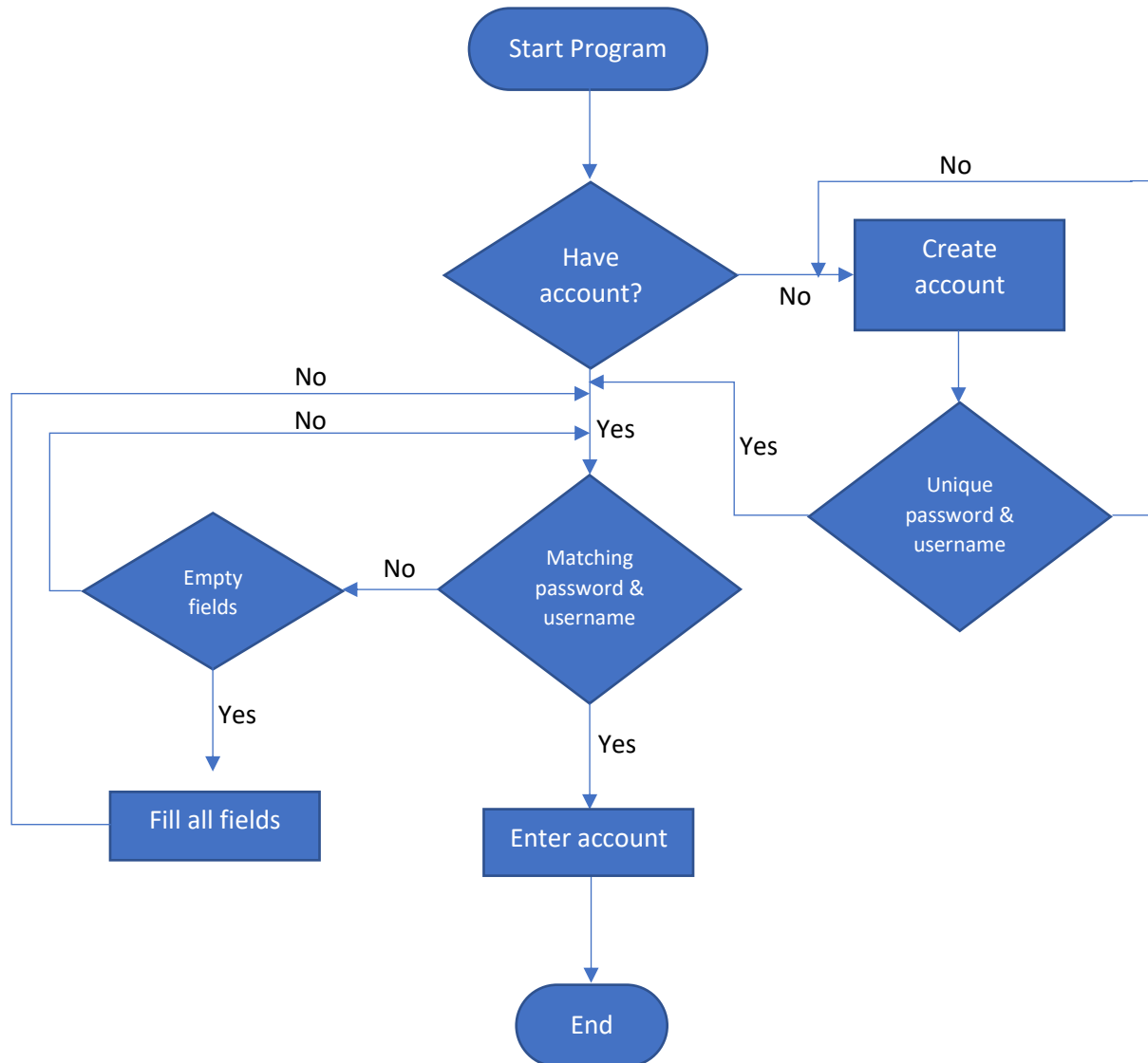


## Criterion B: Design

### Design Plan

The program will consist of several parts: the login / create account interface, the profile interface, the menu, the opportunities display and error messages.

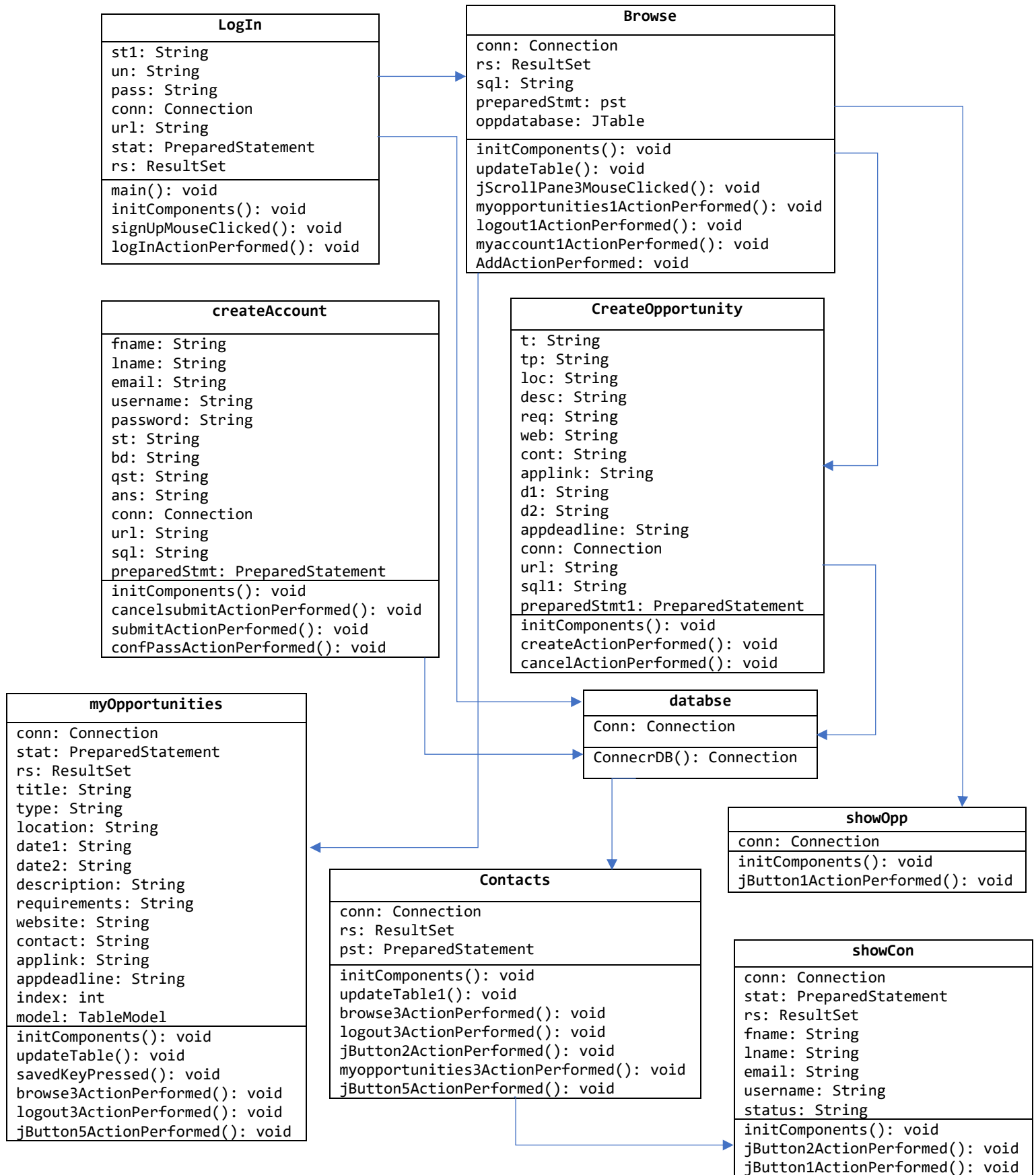
Below is a flowchart that explains how the function of logging in will work.



Several errors that could happen within this function are:

- The password or the username are not correct
- The user does not fill at least one of the boxes

## UML class structure



## Interface Design

To design the interface, I used NetBeans 8.2 and created a JFrame for each window. I used NetBeans 8.2 because it allowed me to drag all the elements needed for the GUI and automatically generate code for them.

Some of the autogenerated code looks like this:

```
private void initComponents() {
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jPanel1 = new javax.swing.JPanel();
    jLabel11 = new javax.swing.JLabel();
    jLabel12 = new javax.swing.JLabel();
    jLabel13 = new javax.swing.JLabel();
    password = new javax.swing.JPasswordField();
    username = new javax.swing.JTextField();
    statusSelect = new javax.swing.JComboBox<>();
    login = new javax.swing.JButton();
    signUp = new javax.swing.JButton();
    jLabel6 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel4.setText("You don't have an account?");

    jLabel5.setFont(new java.awt.Font("Tw Cen MT Condensed Extra Bold", 0, 24)); // NOI18N
    jLabel5.setText("Log In to Opportunity Finder");

    jPanel1.setBackground(new java.awt.Color(200, 200, 200));

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 426, Short.MAX_VALUE)
    );
    jPanel1Layout.setVerticalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 22, Short.MAX_VALUE)
    );

    jLabel11.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
    jLabel11.setText("USERNAME:");

    jLabel12.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
    jLabel12.setText("PASSWORD:");

    jLabel13.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
    jLabel13.setText("STATUS:");

    password.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            passwordActionPerformed(evt);
        }
    });
    password.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            passwordKeyReleased(evt);
        }
    });

    username.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```

        usernameActionPerformed(evt);
    }
});
username.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        usernameKeyReleased(evt);
    }
});

statusSelect.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Select",
"Organization", "Student" }));
statusSelect.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        statusSelectMouseClicked(evt);
    }
});
statusSelect.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        statusSelectActionPerformed(evt);
    }
});

logIn.setText("Log in");
logIn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        logInActionPerformed(evt);
    }
});

signUp.setText("Sign up");
signUp.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        signUpMouseClicked(evt);
    }
});
signUp.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        signUpActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createSequentialGroup()
                    .addGap(80, 80, 80)
                    .addComponent(jLabel5))
                .addGroup(layout.createSequentialGroup()
                    .addGap(80, 80, 80)
                    .addComponent(jLabel11)
                    .addGap(21, 21, 21)
                    .addComponent(username, javax.swing.GroupLayout.PREFERRED_SIZE, 148,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(70, 70, 70)
                    .addComponent(jLabel14)
                    .addGap(4, 4, 4)
                    .addComponent(signUp)
                    .addGap(10, 10, 10)
                    .addComponent(logIn))
                .addGroup(layout.createSequentialGroup()
                    .addGap(80, 80, 80)
                    .addComponent(jLabel13))
            )
        )
    );

```

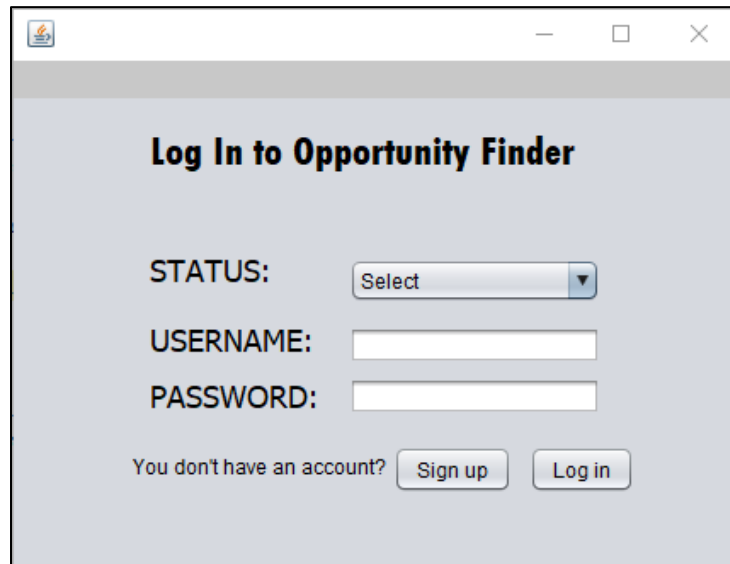
```

        .addGap(46, 46, 46)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel16)
            .addComponent(statusSelect, javax.swing.GroupLayout.PREFERRED_SIZE, 148,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(layout.createSequentialGroup())
            .addGap(80, 80, 80)
            .addComponent(jLabel12)
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel18)
                .addComponent(jLabel17)
                .addComponent(password, javax.swing.GroupLayout.PREFERRED_SIZE, 148,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jLabel15)
            .addGap(46, 46, 46)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel13)
                .addGroup(layout.createSequentialGroup())
                    .addGap(4, 4, 4)
                    .addComponent(statusSelect, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel16)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createSequentialGroup())
                    .addGap(8, 8, 8)
                    .addComponent(username, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(1, 1, 1)
            .addComponent(jLabel17)
            .addGap(3, 3, 3)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel12)
                .addComponent(password, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(2, 2, 2)
            .addComponent(jLabel18)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup())
                    .addGap(4, 4, 4)
                    .addComponent(jLabel14))
                .addComponent(signUp)
                .addComponent(logIn))
            .addContainerGap(138, Short.MAX_VALUE))
    );

    pack();
    setLocationRelativeTo(null);
} // </editor-fold>

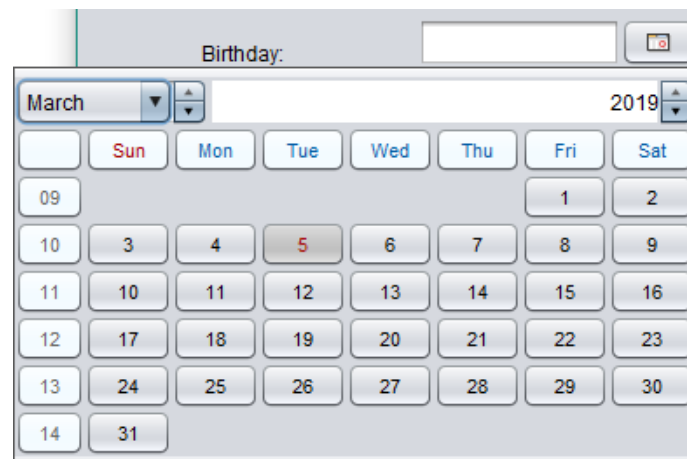
```

The results is this:



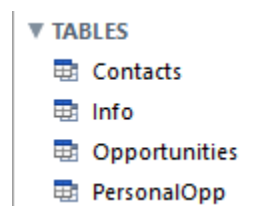
In some cases I import some jar files to my project library to enable me to write several functions. The libraries I use are `jcalendar-1.4` through which I can choose dates, `rs2xml` which enables me to connect a `jTable` with a database, and `sqlitejdbc-v056` which creates the connection with the sql database and allows all statements to be executed.

An example of using the `jCalendar`:



## SQLite Database

To manage the SQL tables I used a software called `SQLiteManager`. There I was able to create and edit the tables I needed for my program.



The `Contacts` table saves the contacts users add to their account.

The `Info` table has all the information of the users when they create an account.

The `Opportunities` table has the details for each opportunity.

The `PersonalOpp` table includes the saved opportunities of the user.

The Info table looks like the image below:

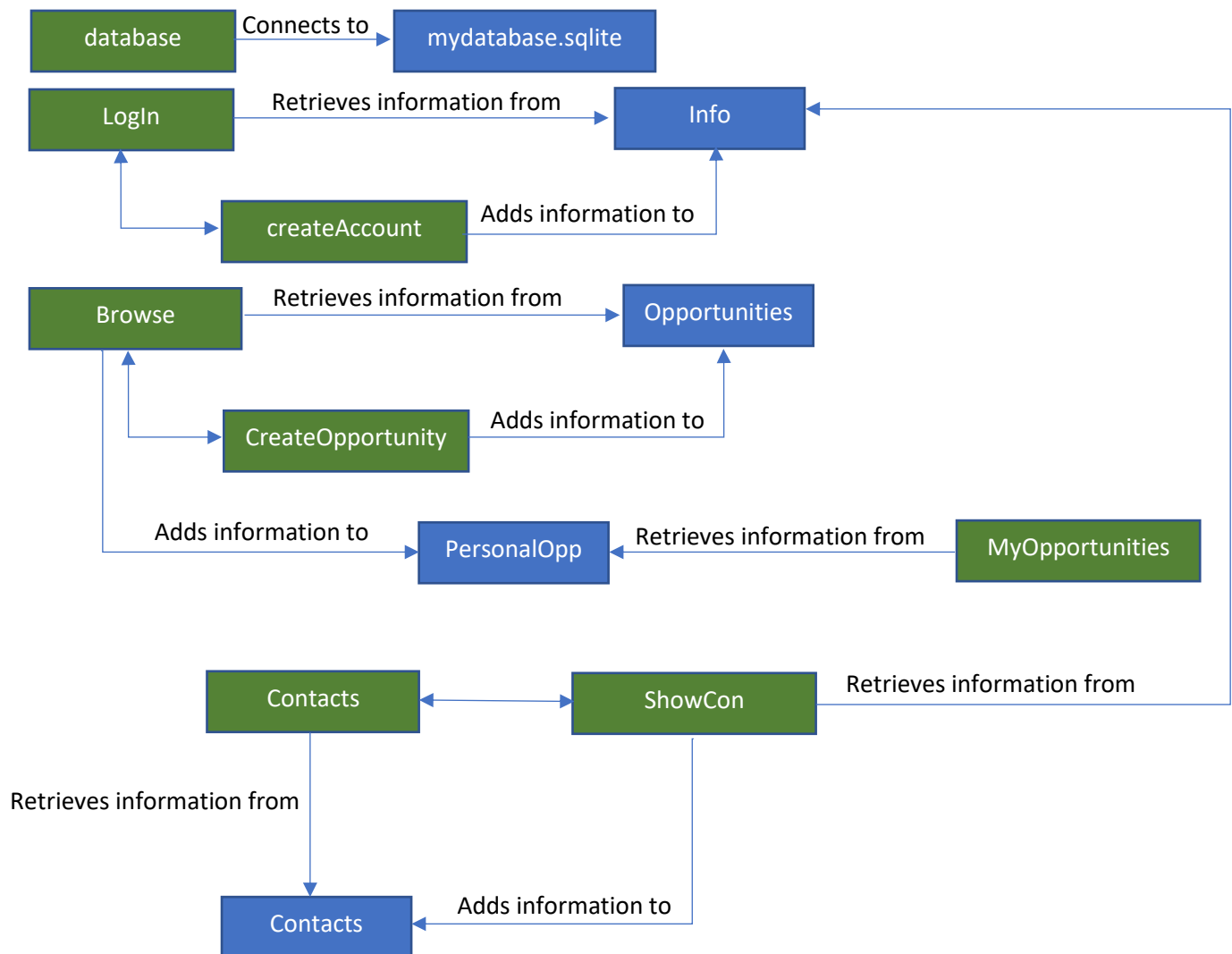
rowid	FirstName	LastName	Email	Username	Password	Status	Birthday	Question	Answer
1	Erlisa	Kulla	erlisa.kulla	erlisa.k	123	Student	NULL	NULL	
2	Ori	Dagan	ori.d@em-is.org	ori.d	123	Select	Mar 3, 2019	Where were you b...	Israel

The Opportunities table looks like the image below:

rowid	Title	Type	Description	Location	Date1	Date2	Requirements	Website	Contact	ApplicationLink	ApplicationD...
1	Internship in T...	Internships	Summer oppo...	Tel Aviv Unive...	2019-01-02	2019-01-05	Resume	www.tau.edu	tau@edu.il	www.applyher...	2019-01-02

In my programs I use several algorithms to add data into the table, copy and paste data to another table, however the removal of data can be done only manually by the admin.

Below is a diagram that shows the connection between classes and tables and between tables themselves. Classes are in green and tables/databases are in blue.



## Test Plan

Action	Success Criteria	Method
Log In	The user will successfully be able to log in if the username and password are correct, otherwise, error messages appear.	<input type="checkbox"/> Write down the username and password or select the wrong status to check that the data it retrieved in the right way. <input type="checkbox"/> Leave one of the fields empty to see if the error message will appear. <input type="checkbox"/> Write down the correct credentials and successfully log in to the program.
Create Account	Users can create an account an login successfully afterwards.	<input type="checkbox"/> Fill in every box with the right type of data. <input type="checkbox"/> After submission check if the information is saved in the database. <input type="checkbox"/> After submission the page redirects to the login page and if the new user credentials are inputted, the user can log in. <input type="checkbox"/> Type a username that already exists and see if the error message appears. <input type="checkbox"/> Check that the cancel button opens the login page and doesn't create any new users.
Add Opportunities		<input type="checkbox"/> Fill in every box with the right type of data. <input type="checkbox"/> After submission check if the information is saved in the database and they appear on the Browse table. <input type="checkbox"/> After submission the page closes. <input type="checkbox"/> Check that the cancel button doesn't create any new opportunities. <input type="checkbox"/> Try to see that an error message appears if any field is empty.
View Opportunities' details	After the selection of one item of the table, a frame with more details appears.	<input type="checkbox"/> Click in one of the items of the table and see that the right details appear on a pop-up frame. <input type="checkbox"/> The Close button closes the frame.
Add opportunities to personal list	The users can add opportunities to the MyOpportunities tab table by choice.	<input type="checkbox"/> When the details frame opens, the user will be able to add it to their list through the Add button. <input type="checkbox"/> Users can remove opportunities from their list.
Display contacts		<input type="checkbox"/> The list of every user appears on a table. <input type="checkbox"/> When clicking in each user, more details appear.
Add contacts to contact list	Users can add other users as their contacts.	<input type="checkbox"/> Each user can add specific people to their list of users. <input type="checkbox"/> Try removing contacts from the list
Log Out	The information is saved after the user logs out from the program and logs in afterwards.	<input type="checkbox"/> After clicking the Log out button, the program opens the Log in page.

Word count: 295