



Roommate Matching System

Senior Project

In partial fulfillment of the requirements for  
The Esther G. Maynor Honors College  
University of North Carolina at Pembroke

By

Lucas Williams  
Computer Science  
May 10, 2021

*Lucas Williams*

May 10, 2021

-----  
Lucas Williams  
Honors College Scholar

-----  
Date

*s.mohanarajah*

May 20, 2021

-----  
Selvarajah Mohanarajah  
Faculty Mentor

-----  
Date

-----  
Joshua Kalin Busman, Ph.D.  
Senior Project Coordinator

-----  
Date

### Acknowledgements

I, Lucas Williams, would like to personally thank professor Selvarajah Mohanarajah, who is also the chair of the department of computer science at UNC Pembroke, for mentoring me throughout the creation process of this project as well as various similarly difficult projects. Thanks also goes to professor Joshua Busman, who oversaw the Honors Thesis/Project class in which this project was completed, as well as being very helpful through several emails requesting assistance.

### Abstract

The senior project I have created was more of a proof of concept for a roommate matching system, where students (UNC Pembroke students with a valid Banner ID were used as an example) would input their personal information and preferences for various different categories into my application, and would be matched with others that input similar preferences in a separate database. This in theory would allow for less conflict among roommates, which can lead to a more fulfilling college experience, including heightened confidence in ability to do well in classes, as well as healthier social experiences with peers.

## Roommate Matching System

In college, roommates are often required for student living spaces so that the cost of living in them is low enough to attract more students. This is achievable by either having students share the same room and lobby, including a bathroom, or by multiple students having single rooms to themselves, but also sharing a lobby that is connected to all of the rooms, including a bathroom. Any living situation that does not include roommates in any way will increase the cost of living by a substantial amount, such as a single apartment room. Roommates can cause issues among themselves and can ruin one's college experience, so it is very important that roommates be compatible to the point where they feel comfortable around each other, and likely to be able to form some sort of friendship or partnership. It can be advantageous for roommates to have similar interests so that these sort of bonds can form naturally.

A piece of software that could match highly compatible roommates with each other would be extremely useful for the living quality of a college campus; this software would ideally be used by every student that would be having a roommate. The software should include a survey that will collect answers of preferences in certain categories such as music taste, hobbies, intended major, and classification (grade). Intended major and student classification will help pair students that will likely have similar paths in college, so that they could possibly help each other and share classes. Hobbies such as sports could help form natural bonds between roommates and spark a conversation, and they could have something physical to do for fun.

Music is often used as a tool to relax or to study, and so similar music taste could help people do these things together, or at least one roommate would not mind the other roommate playing certain music while they focused on schoolwork or just sat back and relaxed. There are many more categories that can be used to minimize conflict between roommates and lead to a much better college experience.

Conflict between roommates can happen often, especially if similar interests are not shared. Simply because one person does not like something, others can end up not liking that person. A personality test could also be used in conjunction with the survey to also see what type of person the user is, such as if they have an introverted or extroverted personality. As an introvert, having an introverted roommate in the same room can often be boring, but at the same time, it can lead to an easier friendship and less moments where friends of the other person come to the room and cause a large mess or annoyance. There are also upsides to having an extroverted roommate as an introvert though, such as being able to meet more people through that person, and this sort of networking can be valuable.

Personal experiences of having roommates in college that were disliked have inspired this idea, and I know how having people around you that do not share the same interests can affect one's daily life. Conflicts such as being loud at certain times of the day or night when one person is studying can occur often if one roommate is not understanding of the others' needs or priorities in life. If random students are matched to be roommates, there is a high possibility that some students will have to room with other students that will not prioritize school work as much

as they do. Partying at night is common on a college campus, which is loud and can interrupt important time students need to either sleep to prepare for classes, or study for a test they might have the following morning. Even if neither of these are the case for the student that is annoyed by the noise, the student might simply want peace and quiet and this needs to be respected.

This project will be an extension of a project theorized in CSC 2920 “Software Development and Professional Practices.” In this class, the main concept of the project was laid out, but not much implementation was done. The software created will be accessible via computer or mobile devices, as it will be web-based. There is the possibility to create an app for this sort of software, but making a web-based version would be most convenient.

At the University of North Carolina at Pembroke, there is no roommate matching system for any of the residence halls or campus apartments. It is simply up to fate that decides whether the student has a roommate that will complement their college experience, or destroy it. Many other universities have some sort of method to pair students together with similar interests to avoid conflict, and it has many benefits concerning the well-being of the student. However, this thinking does not have to stop at just universities; having a system to match people together based on certain criteria is very useful for sports camps, apartments in general, dating apps, and even finding candidates for group projects in class. This has been proven especially by dating apps, where people can more likely find love based on criteria given by the user. There is a need for a roommate matching system at UNC Pembroke because its effect on students’ personal lives and academics would be immense.



## References

SQL Commands. Codecademy. (n.d.). <https://www.codecademy.com/articles/sql-commands/>.

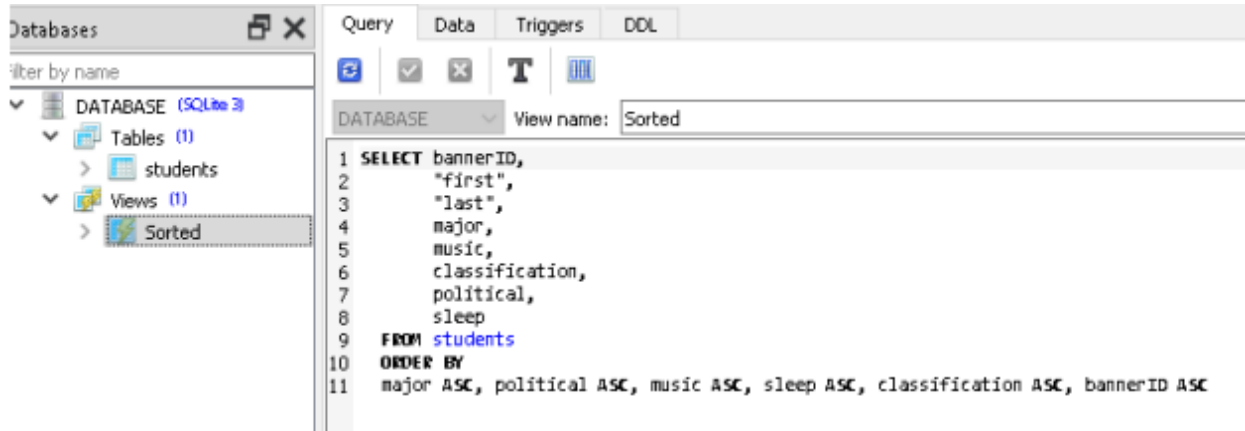
Connect To The SQLite Database Using SQLite JDBC Driver. SQLite Tutorial. (2021, April 10).

<https://www.sqlitetutorial.net/sqlite-java/sqlite-jdbc-driver/>.



## Appendix

I have created an Eclipse IDE Java program that launches an interface and sends information from this form to a database where it is sorted and is a proof-of-concept for a program that could be implemented into any sort of school system or other system that requires people to be roommates.

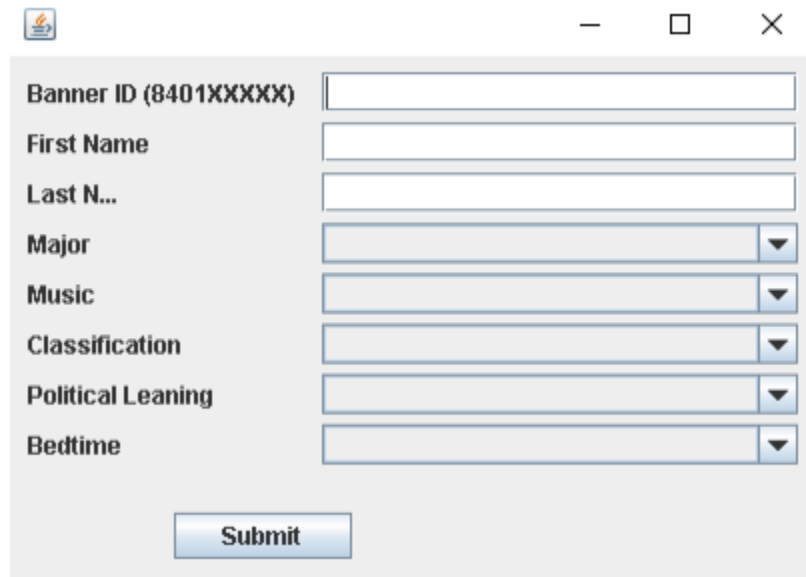


Here is some sample data I inputted from my interface that can be changed based on the user's needs (the current view of the database is sorted by the query above):

	bannerID	first	last	major	music	classificatio	political	sleep
1	6	s	d	Biology	Blues	Freshman	Conservative	Earlier (before midnight)
2	7	s	d	Biology	Blues	Freshman	Conservative	Earlier (before midnight)
3	8	s	d	Biology	Blues	Freshman	Conservative	Earlier (before midnight)
4	9	s	d	Biology	Blues	Freshman	Conservative	Earlier (before midnight)
5	765448	kghf	ljhgfljgh	Biology	Rock	Senior	Conservative	Late (midnight or beyond)
6	765449	kghf	ljhgfljgh	Biology	Rock	Senior	Conservative	Late (midnight or beyond)
7	5	s	d	Biology	Blues	Freshman	Liberal	Earlier (before midnight)
8	765446	kghf	ljhgfljgh	Computer Science	Rock	Senior	Conservative	Late (midnight or beyond)
9	3	s	d	Computer Science	Blues	Freshman	Liberal	Earlier (before midnight)
10	4	s	d	Economics	Blues	Freshman	Liberal	Earlier (before midnight)
11	97	hj	fg	Economics	Rock	Sophomore	Other	Late (midnight or beyond)
12	2	a	a	Psychology	Jazz	Sophomore	Conservative	Earlier (before midnight)
13	840147224	Lucas	Williams	Psychology	Jazz	Freshman	Liberal	Earlier (before midnight)

The data has the primary key as the BannerID so that duplicate students cannot exist.

As shown above, the filters are at the top. Here is the interface when my program is launched:



The screenshot shows a Java Swing window with a light gray background. At the top left is a small icon, and at the top right are standard window controls (minimize, maximize, close). The main area contains a form with the following elements:

- Banner ID (8401XXXXX)**: A text input field.
- First Name**: A text input field.
- Last N...**: A text input field.
- Major**: A dropdown menu with a downward arrow.
- Music**: A dropdown menu with a downward arrow.
- Classification**: A dropdown menu with a downward arrow.
- Political Leaning**: A dropdown menu with a downward arrow.
- Bedtime**: A dropdown menu with a downward arrow.
- Submit**: A button located at the bottom left of the form area.

The drop-down boxes show all the possible categories that can be used to group people together.

Found below is the source code for the Java program (Eclipse IDE was used):

### **Class (windowbuilder): GUI:**

```
package project;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JLabel;

import java.awt.BorderLayout;

import javax.swing.JTextField;

import javax.swing.JComboBox;

import javax.swing.JSeparator;

import javax.swing.JFormattedTextField;

import javax.swing.JButton;

import java.awt.event.ActionListener;
```

## 11 Williams

```
import java.awt.event.ActionEvent;
```

```
public class GUI extends javax.swing.JFrame {
```

```
    private JFrame frame;
```

```
    private JTextField textField_1;
```

```
    private JTextField textField_2;
```

```
    private JTextField textField;
```

```
    /**
```

```
     * Launch the application.
```

```
     */
```

```
    public static void main(String[] args) {
```

```
        EventQueue.invokeLater(new Runnable() {
```

```
            public void run() {
```

```
                try {
```

```
                    GUI window = new GUI();
```

```
                    window.frame.setVisible(true);
```

```
                } catch (Exception e) {
```

```
                    e.printStackTrace();
```

```
                }
```

```
            }
```

```
        });
```

```
    }
```

```
    /**
```

```
     * Create the application.
```

```
     */
```

```
    public GUI() {
```

```
        initialize();
```

```
    }
```

```
/**
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 421, 303);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JLabel lblNewLabel = new JLabel("Banner ID (8401XXXXXX)");
    lblNewLabel.setBounds(10, 11, 137, 14);
    frame.getContentPane().add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("First Name");
    lblNewLabel_1.setBounds(10, 36, 125, 14);
    frame.getContentPane().add(lblNewLabel_1);

    JTextField textField_1 = new JTextField();
    textField_1.setBounds(157, 33, 238, 20);
    frame.getContentPane().add(textField_1);
    textField_1.setColumns(10);

    JLabel lblNewLabel_2 = new JLabel("Last Name");
    lblNewLabel_2.setBounds(10, 61, 46, 14);
    frame.getContentPane().add(lblNewLabel_2);

    JLabel lblNewLabel_3 = new JLabel("Major");
    lblNewLabel_3.setBounds(10, 86, 46, 14);
    frame.getContentPane().add(lblNewLabel_3);
}
```

## 13 Williams

```
textField_2 = new JTextField();  
textField_2.setBounds(157, 58, 238, 20);  
frame.getContentPane().add(textField_2);  
textField_2.setColumns(10);
```

```
JLabel lblNewLabel_4 = new JLabel("Music");  
lblNewLabel_4.setBounds(10, 111, 46, 14);  
frame.getContentPane().add(lblNewLabel_4);
```

```
JLabel lblNewLabel_5 = new JLabel("Classification");  
lblNewLabel_5.setBounds(10, 136, 95, 14);  
frame.getContentPane().add(lblNewLabel_5);
```

```
JLabel lblNewLabel_6 = new JLabel("Political Leaning");  
lblNewLabel_6.setBounds(10, 161, 95, 14);  
frame.getContentPane().add(lblNewLabel_6);
```

```
JLabel lblNewLabel_7 = new JLabel("Bedtime");  
lblNewLabel_7.setBounds(10, 186, 95, 14);  
frame.getContentPane().add(lblNewLabel_7);
```

```
String majors[] = { null, "Psychology", "Economics", "Computer Science", "Biology", "Engineering",  
"Education", "Business", "Nursing", "Other" };
```

```
JComboBox comboBox1 = new JComboBox(majors);  
comboBox1.setBounds(157, 83, 238, 20);  
frame.getContentPane().add(comboBox1);
```

```
String musics[] = { null, "Jazz", "Blues", "Rock", "R&B", "Metal", "Hip Hop", "Country", "Pop", "Other" };
```

```
JComboBox comboBox2 = new JComboBox(musics);
```

```
comboBox2.setBounds(157, 108, 238, 20);  
frame.getContentPane().add(comboBox2);  
  
String classifications[] = { null, "Freshman", "Sophomore", "Junior", "Senior" };  
  
JComboBox comboBox3 = new JComboBox(classifications);  
comboBox3.setBounds(157, 133, 238, 20);  
frame.getContentPane().add(comboBox3);  
  
String politics[] = { null, "Liberal", "Conservative", "Moderate", "Libertarian", "Other", "Not Affiliated" };  
  
JComboBox comboBox4 = new JComboBox(politics);  
comboBox4.setBounds(157, 158, 238, 20);  
frame.getContentPane().add(comboBox4);  
  
String sleeping[] = { null, "Earlier (before midnight)", "Late (midnight or beyond)" };  
JComboBox comboBox5 = new JComboBox(sleeping);  
comboBox5.setBounds(157, 183, 238, 20);  
frame.getContentPane().add(comboBox5);  
textField = new JTextField();  
textField.setBounds(157, 8, 238, 20);  
frame.getContentPane().add(textField);  
textField.setColumns(10);  
JButton btnNewButton = new JButton("Submit");  
btnNewButton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        int bannerID = Integer.valueOf(textField.getText());  
        String first = textField_1.getText();  
        String last = textField_2.getText();  
        String major = comboBox1.getSelectedItem().toString();  
        String music = comboBox2.getSelectedItem().toString();  
        String classification = comboBox3.getSelectedItem().toString();
```

```

        String political = comboBox4.getSelectedItemAt().toString();

        String bedtime = comboBox5.getSelectedItemAt().toString();

        matching app = new matching();

        // insert three new rows

        app.insert(bannerID, first, last, major, music, classification, political, bedtime);

        }

    });

    btnNewButton.setBounds(83, 227, 89, 23);

    frame.getContentPane().add(btnNewButton);

}
}

```

**Class: matching:**

```

package project;

import java.sql.*;

public class matching {

    private Connection connect() {

        // SQLite connection string

        String url = "jdbc:sqlite:C:\\Users\\Luke\\eclipse-workspace\\Senior Project\\DATABASE.db";

        Connection conn = null;

        try {

            conn = DriverManager.getConnection(url);

        } catch (SQLException e) {

```

## 16 Williams

```
        System.out.println(e.getMessage());
    }
    return conn;
}

public void insert(int bannerID, String first, String last, String major, String music, String classification, String political, String
sleep) {
    String sql = "INSERT INTO students(bannerID,first,last,major,music,classification,political,sleep)
VALUES(?,?,?,?,?,?,?)";

    try (Connection conn = this.connect();
        PreparedStatement pstmt = conn.prepareStatement(sql)) {
        pstmt.setInt(1, bannerID);
        pstmt.setString(2, first);
        pstmt.setString(3, last);
        pstmt.setString(4, major);
        pstmt.setString(5, music);
        pstmt.setString(6, classification);
        pstmt.setString(7, political);
        pstmt.setString(8, sleep);

        pstmt.executeUpdate();
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
}

public static void main(String[] args) {
}
```



}

**The SQLite database included a view which was sorted using the query:**

```
SELECT bannerID,  
"first",  
"last",  
major,  
music,  
classification,  
political,  
sleep  
FROM students  
ORDER BY  
major ASC, political ASC, music ASC, sleep ASC, classification ASC, bannerID ASC
```