**Global Student Network Generator**

**Members**

**Rita Chen - 1359214**

**Aron Lin - 1348828**

**Htoo Naing - 1350488**

**Zhasmin Tuiachieva - 1342449**

**Computer Programming II – CSCI-185 – Section M03**

**Table of Contents**

Abstract ........................................................................................................................................ 3

Introduction .................................................................................................................................. 4

Technical Specifications ............................................................................................................... 5

Diagrams ..................................................................................................................................... 11

Project Results ............................................................................................................................ 13

Outline ......................................................................................................................................... 13

Components .................................................................................................................... 13

Main Page ........................................................................................................... 13

Instructions Page ................................................................................................. 14

Network Page ...................................................................................................... 14

Map Panel ........................................................................................................... 14

University Dropdown Panel ................................................................................ 15

File Manager ....................................................................................................... 15

Connection Strength Bar ..................................................................................... 15

Person, Student, University Classes .................................................................... 16

Problems Faced ............................................................................................................... 16

Showcase ..................................................................................................................................... 19

Contributions ............................................................................................................................... 20

Conclusion .................................................................................................................................. 23

**Abstract**

International students typically find trouble integrating into campus life due to feelings relating to cultural norms and general anxiety. This can often lead to many foreign undergraduates being unable to establish their own social circles, reducing the interactions they have with like-minded peers during college and potentially causing trouble down the line for their mental health. This issue can be remedied by providing these students with access to a resource that allows them to connect with similar international undergraduates from around the world, bridging the gap between college communities and foreign students.

The global student network generator is an application specifically designed for foreign undergraduates that attempts to solve this problem. By enabling foreign undergraduates to connect with others around the world through a international student network, they are provided with social circles which they can join and socialize in. This is done through allowing individual users to create their own student accounts within a world-wide simulation, setting up profiles based on chosen interests, along with the universities they attend and the college courses they are undertaking.

The application works through multiple GUI windows which have the premise of allowing a user to create a student profile. Users then have their profiles displayed across the simulation, enabling other students to connect to them based on a given connection strength score. This score is dependent on the likeness of traits or college careers of each student’s profile, with the most positive impact to the strength integer being in the same university location.

**Introduction**

International students often have the difficulty of feeling uncomfortable and unknown to many people when leaving their home countries to arrive on campus for the first time. As a result, they often tend to not reach out to peers and can have problems making friends on campus to associate with. By introducing a linear system in which international students can provide their interests, majors, or universities, we can develop an application which sets up connections between students based on the number of similarities found.

With this stream-lined and easily accessible method, we grant many international students the ability to meet with peers from other countries who are also undergoing the same feeling of ostracization, bridging a gap between college communities and foreign undergraduates based on the premises of similarity. Moreover, students are matched in accordance to how “similar” each one’s interests and college career currently are – ensuring that matches are kept to being as relevant as possible to one another. This essentially allows international students to find one another based on how relevant one’s interests and/or college career are, acting as a pseudo-LinkedIn based on a person’s profile.

**Technical Specifications**

[1] “HashMap in Java with Examples,” *GeeksforGeeks*, Apr. 28, 2017. <https://www.geeksforgeeks.org/java-util-hashmap-in-java-with-examples/>

[2] “Java: create a list of HashMaps,” *Stack Overflow*. <https://stackoverflow.com/questions/4100486/java-create-a-list-of-hashmaps>

[3] “Stack Overflow,” *Stack Overflow*, Nov. 13, 2011. <https://stackoverflow.com/questions/8114406/adding-image-to-jpanel-through-imageio-read>

[4] Overriding paintComponent, “Stack Overflow,” *Stack Overflow*, Mar. 24, 2013. <https://stackoverflow.com/questions/15602235/overriding-paintcomponent>

[5] “Graphics (Java Platform SE 8),” *Oracle.com*, Apr. 05, 2025. <https://docs.oracle.com/javase/8/docs/api/java/awt/Graphics.html#drawImage>

[6] “Graphics2D (Java Platform SE 8),” *docs.oracle.com*. <https://docs.oracle.com/javase/8/docs/api/java/awt/Graphics2D.html>

[6] “paint() and repaint() in Java,” *Stack Overflow*, May 26, 2012. <https://stackoverflow.com/questions/10768619/paint-and-repaint-in-java>

[7] “How can I use a custom font in Java?” *Stack Overflow*. <https://stackoverflow.com/questions/5652344/how-can-i-use-a-custom-font-in-java>

‌ [8] “Stack Overflow,” *Stack Overflow*, Oct. 04, 2019. <https://stackoverflow.com/questions/58240025/how-to-draw-any-text-using-drawstring-in-java>

[9] “Java 8 | Consumer Interface in Java with Examples,” *GeeksforGeeks*, Sep. 24, 2018. <https://www.geeksforgeeks.org/java-8-consumer-interface-in-java-with-examples/>

[10] “Consumer (Java Platform SE 8),” *docs.oracle.com*. <https://docs.oracle.com/javase/8/docs/api/java/util/function/Consumer.html>

[11] “How to Write a Mouse Listener (The JavaTM Tutorials > Creating a GUI With JFC/Swing > Writing Event Listeners),” *docs.oracle.com*. <https://docs.oracle.com/javase/tutorial/uiswing/events/mouselistener.html>

[12] “MouseListener and MouseMotionListener in Java,” *GeeksforGeeks*, May 02, 2018. <https://www.geeksforgeeks.org/mouselistener-mousemotionlistener-java/>

[13] “Java MouseListener,” *Stack Overflow*. <https://stackoverflow.com/questions/2668718/java-mouselistener>

[14] Oracle, “List (Java Platform SE 8),” *docs.oracle.com*. <https://docs.oracle.com/javase/8/docs/api/java/util/List.html>

[15] “Rectangle (Java Platform SE 8),” *Oracle.com*, Dec. 04, 2024. <https://docs.oracle.com/javase/8/docs/api/java/awt/Rectangle.html>

[16] “Detecting a mouse click inside a box painted in Frame? [Solved] (Swing / AWT / SWT forum at Coderanch),” *Coderanch.com*, 2025. <https://coderanch.com/t/645861/java/Detecting-mouse-click-box-painted>

[17] “How to Use GridBagLayout (The JavaTM Tutorials > Creating a GUI With JFC/Swing > Laying Out Components Within a Container),” *Oracle.com*, 2017. <https://docs.oracle.com/javase/tutorial/uiswing/layout/gridbag.html>

[18] “Color (Java Platform SE 7 ),” *docs.oracle.com*. <https://docs.oracle.com/javase/7/docs/api/java/awt/Color.html>

[19] “JProgressBar (Java Platform SE 8 ),” *Oracle.com*, Dec. 04, 2024. <https://docs.oracle.com/javase/8/docs/api/javax/swing/JProgressBar.html>

‌ [20] “How to set an image as a background for Frame in Swing GUI of java?,” *Stack Overflow*, Sep. 23, 2009. <https://stackoverflow.com/questions/1466240/how-to-set-an-image-as-a-background-for-frame-in-swing-gui-of-javaound-for-frame-in-swing-gui-of-java>

[21] “Serialization and Deserialization in Java with Example” *GeeksforGeeks*, Feb. 22, 2019. <https://www.geeksforgeeks.org/serialization-in-java/>

‌ [22] “How to Align Text in HTML?,” *GeeksforGeeks*, Jun. 20, 2021. <https://www.geeksforgeeks.org/how-to-align-text-in-html/>

‌ [23] “How to center the text in a JLabel?,” *Stack Overflow*, Jul. 25, 2011. <https://stackoverflow.com/questions/6810581/how-to-center-the-text-in-a-jlabel>

‌ [24] “Git Commit,” *W3schools.com*, 2021. <https://www.w3schools.com/git/git_commit.asp>

‌ [25] “Convert List to Array in Java,” *GeeksforGeeks*, Mar. 27, 2018. <https://www.geeksforgeeks.org/convert-list-to-array-in-java/> (accessed May 12, 2025).

‌ [26] “ArrayList in a JList,” *Oracle Forums*, Aug. 04, 2008. <https://forums.oracle.com/ords/apexds/post/arraylist-in-a-jlist-8029> (accessed May 12, 2025).

‌ [27] “Lesson: Writing Event Listeners (The JavaTM Tutorials > Creating a GUI With Swing),” *docs.oracle.com*. <https://docs.oracle.com/javase/tutorial/uiswing/events/index.html>

‌ [28] “Scan other application for input in X window,” *Stack Overflow*, Dec. 22, 2014. <https://stackoverflow.com/questions/27595139/swing-how-do-i-detect-which-component-i-clicked> (accessed May 12, 2025).

‌ [29] “A Visual Guide to Layout Managers (The JavaTM Tutorials > Creating a GUI With JFC/Swing > Laying Out Components Within a Container),” *docs.oracle.com*. <https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>

‌ [30] “Introduction to Java Serialization,” *Baeldung*, May 15, 2017. <https://www.baeldung.com/java-serialization>

‌ [31] “Convert List to Array in Java,” *GeeksforGeeks*, Mar. 27, 2018. <https://www.geeksforgeeks.org/convert-list-to-array-in-java/>

‌ [32] “Stack Overflow,” *Stack Overflow*, Oct. 26, 2012. <https://stackoverflow.com/questions/13093410/how-to-detect-mouse-clicks-on-a-specific-object-in-java> (accessed May 12, 2025).

‌ [33] “Stroking and Filling Graphics Primitives (The JavaTM Tutorials > 2D Graphics > Working with Geometry),” *docs.oracle.com*. <https://docs.oracle.com/javase/tutorial/2d/geometry/strokeandfill.html>

‌ [34] Ouko Omondi, “How to resolve “The requested URL returned error:403 in GitHub,” *Medium*, Jan. 15, 2023. <https://medium.com/@akwando15/how-to-resolve-the-requested-url-returned-error-403-in-github-cd66bac6d9fa>

‌ [35] “Gitignore Explained: What is Gitignore and How to Add it to Your Repo,” *freeCodeCamp.org*, Jan. 03, 2020. <https://www.freecodecamp.org/news/gitignore-what-is-it-and-how-to-add-to-repo/>

‌ [36] Coding Examples, “Multi Select JList | Java Swing Tutorial #23,” *YouTube*, Jan. 28, 2021. <https://www.youtube.com/watch?v=WiX3n2BMGIc> (accessed May 12, 2025).

‌ [37] “Font (Java Platform SE 8 ),” *docs.oracle.com*. <https://docs.oracle.com/javase/8/docs/api/java/awt/Font.html>

‌ [38] “BorderLayout (Java Platform SE 7 ),” *docs.oracle.com*. <https://docs.oracle.com/javase/7/docs/api/java/awt/BorderLayout.html>

‌ [39] “JComboBox (Java Platform SE 8 ),” *docs.oracle.com*. <https://docs.oracle.com/javase/8/docs/api/javax/swing/JComboBox.html>

‌ [40] Lazic B., “Java swing GUI tutorial #20: GridBagLayout,” *YouTube*, Apr. 03, 2013. <https://www.youtube.com/watch?v=sq46PYdW4c8> (accessed May 12, 2025).

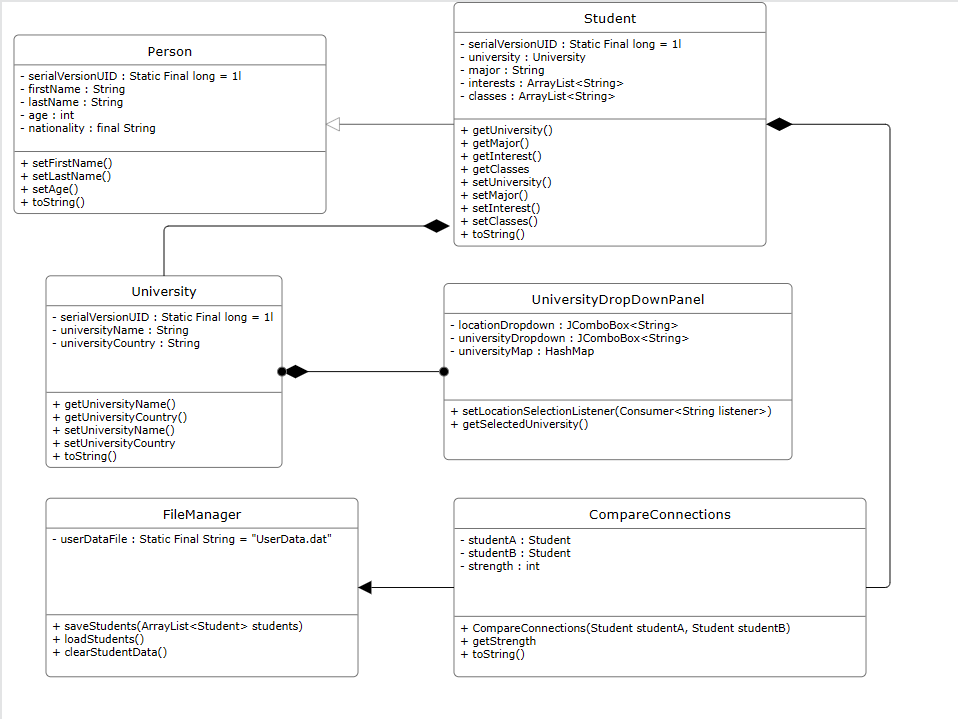
‌ [41] Mam Panhavuth Programming, “How to customize JPanel with GridBagLayout and load Images from Files in Java Using Netbeans IDE,” *YouTube*, May 02, 2020. <https://www.youtube.com/watch?v=mftMp33J4lg> (accessed May 12, 2025).

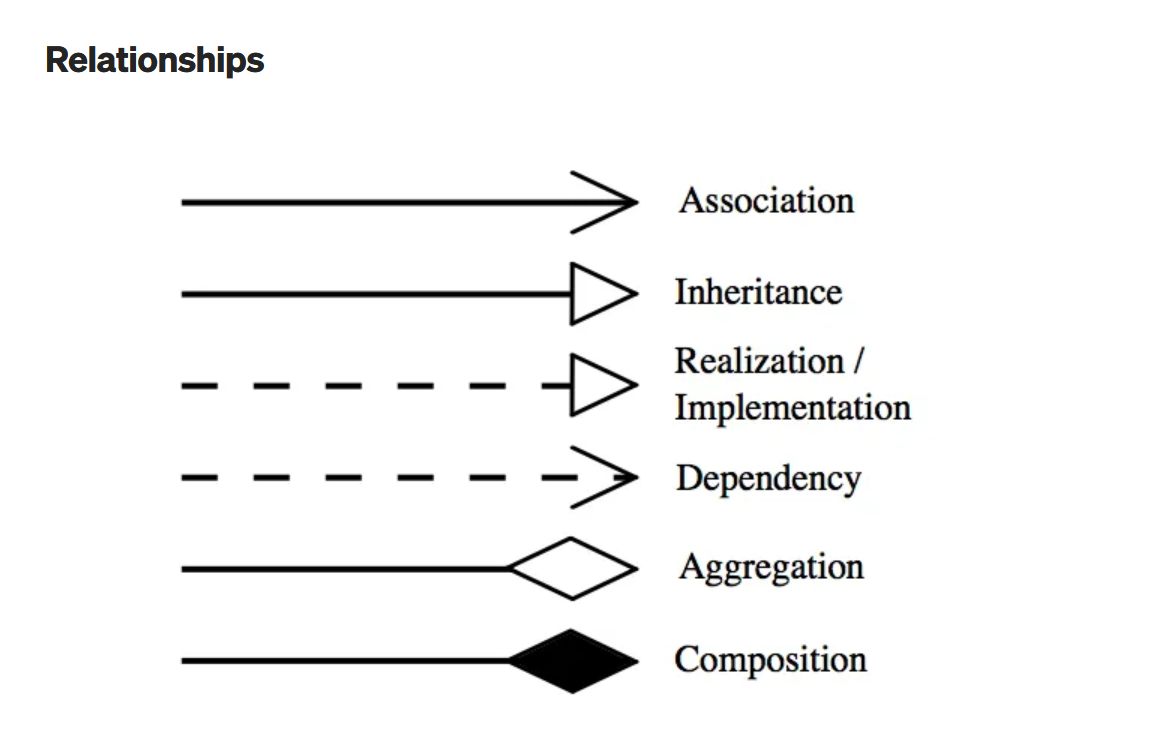
‌ [42] “JFrame | Java Swing Tutorial for Beginners,” *Youtube*, Oct. 19th, 2020. <https://www.youtube.com/watch?v=4BRUmU-ETRk>

‌ [43] “GridLayout | Java Swing Tutorial for Beginners,” *Youtube*, Dec. 29th, 2020. <https://www.youtube.com/watch?v=impJtkTcQ94> (accessed May 03, 2024).

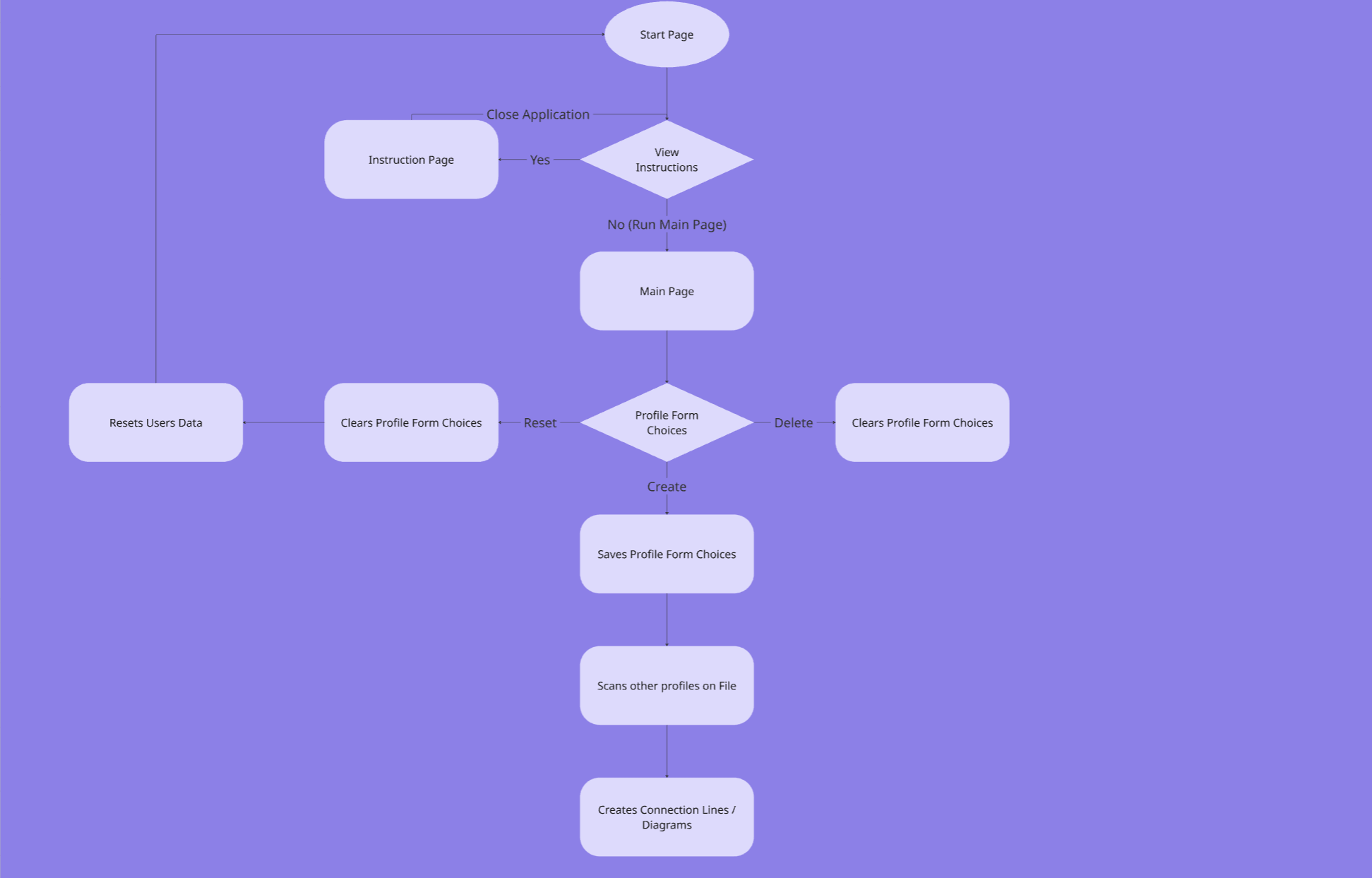
‌ [44] Pritam Thing, “How To Transform One Frame To Another Frame in Java Netbeans,” *YouTube*, Aug. 19, 2018. <https://www.youtube.com/watch?v=PuCxkTcutJI> (accessed May 12, 2025).

**Diagrams**

*UML Diagram (fig. 1)*

[[1]](#footnote-11427)

*UML Diagram Legend (fig. 1a)*

*Flow Diagram (fig. 2)*

**Project Results**

**Outline**

The “global student network generator” is a GUI-based application which aims to solve a common problem with international students concerning their integration into social circles within college life. It acts as a resource for these undergraduates by providing them with the ability to interact and connect with similar students around the world while simultaneously establishing world-wide social circles.

The project utilizes a variety of GUIs to implement the application’s main functionality of an international student network. Many of the main windows within the program are comprised of at least one or two external GUI classes which add onto the functionality of the program. These external GUI classes are often based upon three other classes which define the Person, Student, and University objects.

**Components**

**Main Page**

The main page component serves as the title and general starting point of the application. It provides users with the ability to open the Instructions GUI or the student network page. This component primarily utilizes Java concepts relating to responding to user inputs through ActionListeners. This enables users to progress to the instructions page and network page by clicking on their respective buttons.

**Instructions Page**

The instructions page is a GUI component from the main page. It serves as an interface to guide users in utilizing the application, giving a brief description of each button’s functionality. It utilizes Java concepts relating to basic GUIs, primarily JScrollPane and JTextArea to enable a simple scrolling function and the display of multiple text lines.

**Network Page**

The network page is another GUI component from the main page, acting as one of the two windows that a user can potentially open when starting up the application for the first time. It is the base window for the student network as it houses other GUIs responsible for the main functionality of the application. It utilizes the basic JPanel and ListSelectionListener concept in Java to organize two external panels (University Dropdown & Map) while simultaneously connecting functionality between the pair.

**Map Panel**

The map panel serves as a fundamental component within the application as it implements a world map, along with various functionalities for its pins which represent student profiles. This component utilizes HashMap, paintComponent, MouseListener, and MouseEvent in order to do a variety of things. The HashMap was used to simulate the general coordinates in which a country would be placed, while paintComponent enabled the coloring of each individual pin for student profiles. The MouseListener and MouseEvent methods were utilized to initialize the map while simultaneously giving functionality to pin clicking, allowing the class to expand the selected student profile to give information about the potential connection strength they may have with the user.

**University Dropdown Panel**

The university dropdown panel is one of the two GUIs utilized within the network page, staging as the far-left panel in which the user can apply a sample university from different locations internationally to their student profile. It utilizes the List and ListSelectionListener to provide sample universities in various locations while additionally adding the selected university with its location to a student’s profile.

**File Manager**

The internal file manager is a special and essential component responsible for saving the profiles of users as data to be called back by many of the other components listed here. It utilizes Java concepts like ObjectOutputStream, FileOutputStream, and Try-Catch for both data and error handling functionalities. Each function within the component primarily utilizes the Try-Catch method to handle any potential errors during the data saving or erasing of user profiles, which is handled by both the ObjectOutputStream and FileOutputStream methods respectively.

**Connection Strength Bar**

The connection strength bar is a component responsible for displaying the strength integer (determined off the number of similarities found, through the CompareConnections class) as a graphical bar percentage. It utilizes the GridBagLayout and Color method to implement a specific position for the colored bar graph and its corresponding label. This component additionally utilizes Encapsulation, one of the core OOP principles to ensure flexibility and modularity throughout the class, enabling other classes to easily call functions from the component while simultaneously making it easier to read for anybody who wishes to modify it.

**Person, Student, University Classes**

These classes function more as internal components to the project and act as the bases which many of the GUI components work from. They utilize basic constructors, getters, and setters to allow the classes to construct, obtain, and set many of the objects required within the application. These classes are regularly called by other components within the program and mainly serve as the base internal objects of the application.

**Problems Faced**

The initial problems we faced when setting up the project was deciding what the project was originally going to be on, with team members originally being split between a global student network simulator and a life simulation game. The global student network generator is essentially the project you have before you now, a GUI-based application where international students are able to find one another based on provided interests and college careers – whereas a life simulation game was based on being a random event-driven simulator in which the user makes choices for a character. Both options were completely different categorically, as the global student network generator focused on utility, contrast to the life simulation game’s focus on user entertainment. A compromise was then offered merging the two ideas together, conceptualizing a project wherein users would be able to connect with other users based on a variety of options that they had chosen in the life simulation game. However, this compromise was decidedly scrapped by all group members as they realized fitting this into an approximate two-week timeframe would be very difficult to do. Moreover, the project idea pertaining to the global student network simulator was much simpler than the latter option, and additionally provided more utility or usage compared to a game simulation. This ultimately led to the global student network generator being the final project idea that all the group members had settled upon.

After settling on a project, the next roadblock we had ahead of us was finding the proper documentation and know-how's on performing the tasks we needed to complete for the project. Each member did their research in their own time, with each person’s tasks being varied and requiring a different amount of documentation to complete their part successfully. This process was often done while code was being written, with team members looking for ways to implement specific portions of their work that they previously had no experience with in-class by searching the internet and stumbling upon sites like Stack Overflow or Oracle documentation.

Throughout the project, there were multiple communication mishaps in which members of the group were confused and scrambled about doing their specific tasks. Since each group member’s portion required another member’s portion to be present for valid testing, this led to an issue where a member produced classes originally meant to be done by other group members. Members were additionally confused about specific instructions and variables throughout the project as they seemed vague or had certain plans which were changed by members without prior notice.

There were additionally constant problems concerning the GitHub repository’s main branch – in which a multitude of merge conflicts occurred from attempting to integrate the work of other group members. These conflicts were all handled by one group member, who singlehandedly ensured that compatibility between each member’s portion of work would be able to be implemented seamlessly into the project, ensuring that it would work as intended.

Finally, problems concerning the project’s two-week time constraint appeared constantly during development. There were quite a few features lost from the original design that had been scrapped, primarily due to us being unable to fit it into this two-week timespan. Moreover, group members were stressed and had to make some personal sacrifices during the project in order to get their portions implemented and working correctly in time to meet a minimum product expectation.

**Showcase**

[](https://youtu.be/oZV9fMl1vlw)

**GitHub**

<https://github.com/TRGGB2/GlobalStudentNetworkSimulator/>

**Contributions**

**Rita Chen – 1359214**

* Constructed the original ‘University’ class which contained the name and location fields of sample universities for the project simulation.
* Designed the Map panel class, a notable component which contains the map image, the functions to draw and interact with pins on specific coordinate locations as buttons.
* Represented the geographic location of each student profile utilizing a map image to allow for global user representation.
* Catalogued countries within the map image with numerous sample universities to choose from, enabling usage for student profiles.
* Developed the ‘UniversityDropdownPanel’ class which allowed users to input a country to narrow down the locations that their university is set in.
* Implemented functionality within the map image to allow student profiles to be clicked upon, expanding their information and showing their similarities with a user.

**Aron Lin – 1348828**

* Organized and managed the GitHub repository wherein members would make their coding contributions throughout the entirety of the project, additionally handling all merging to the main branch.
* Established a Google Slide for the presentation, providing a base for the infographics and slides necessary to showcase the project.
* Produced both the UML and Flow Diagram figures.
* Implemented the instructional and main menu GUI portions of the application, setting starting points for the program when launched.
* Produced a video demonstration of the project, walking through the general setup of the code as well as the application.
* Provided graphics and assistance with the presentation slides.

**Htoo Naing – 1350488**

* Composed the project documentation in full on the terms of writing – effectively documenting the entire project within a well-organized and formatted document.
* Oversaw the revision of the document by providing group members with consistent updates to the form, handling feedback and making changes as necessary.
* Drafted and revised the presentation slides in response to group feedback and coordination to ensure optimal publication.
* Originally was drafted to create the ‘ConnectionEngine’, a class which would compare profiles of students and enable connection strength measuring from its similarities.
* Produced a Connection Strength Bar GUI which allowed for the graphical visualization of a student pairing’s connection strength. This GUI responded to certain numerical outputs by changing colors of the bar graphic in accordance with the connection strength integer.

**Zhasmin Tuiachieva – 1342449**

* Proposed and outlined three final project proposals, with an in-depth breakdown and explanation of what each design would entail in terms of methodology and use.
* Established a Final Project Plan which outlined the specific classes and responsibilities of the entire team – along with splitting the tasks between four members.
* Drafted visual mockups of what the project’s output should look like – effectively outlining the final designs of the GUIs in a visual format.
* Prepared the fundamental base ‘Person’ and ‘Student’ classes.
* Modeled a GUI wherein users would input their data in terms of names, age, nationalities, universities, and majors.
* Modified the original ‘University’ class to enable data-saving functionality using a Serializable interface.
* Produced the ‘CompareConnections’ class which compared traits given in the ‘Student’ and ‘Person’ classes to provide a connection strength integer.
* Utilized File I/O to persist student data within a .dat format within the ‘FileManager’ class.

**Conclusion**

This project had plenty of scrapped features during planning and development, primarily due to constraints with the Java language and time. We accomplished a lot of the main functionalities of the program in terms of implementing a student network and allowing users to create their own profile within the application, along with seeing potential connections to their subsequent profiles. When we originally started designing the project, we had plenty of functions in mind that we wanted to implement for the student network – mainly concerning additional functions which would serve to make the project more of a simulation with its ability to produce random events (emulating users changing countries or universities) and compare connections with multiple people at once. However, these features were eventually scrapped due to time constraints, forcing group members to reconsider valuable components of the project in order to deliver a “minimum viable product” (MVP) within the deadline. Throughout this project, we learned to utilize various resources at our disposal concerning programming, mainly the use of coding forums and documentations in order to provide the knowledge required to complete some of the tasks given to us in the project. Moreover, group members learned to organize and use GitHub actively as a means of integrating their work together, providing experience with a tool that some of us had never used extensively in the past.

In the future, we’d like to implement more functions to the project to fit our original description of being a “global student network generator”, namely with the features we originally had to scrap due to time constraints. This would allow us to deliver a potentially powerful application which international undergraduates could utilize on a regular basis to form social connections with similar students, expanding the project’s original audience.

1. [↑](#footnote-ref-11427)