Team: Jamal Farraj, Henry Rivere CSCI 377
9/22/25 Project Topic

For this project, our team decided to focus on the Banking Management System where we intend to design and create a system that mimics everyday banking transactions such as creating an account, checking your balance, the deposit and withdrawal from bank accounts and even the transfer of funds between different accounts. We will also focus on how different algorithms and data structures can help or hinder our progress and results when it comes to maintaining all account information. By using search and sort algorithms to store account information, we can see how the system performance is affected and compare it to other ways of storing account information.

The system will be built using an Account class which stores the users account number, name, and account balance. Users will also have the ability to make deposits, withdrawals and other regular transactions. To implement these features in a proper working capacity, account information will be stored in a container data structure so that they may be searched for easier. We plan to use a sorted std::vector<Account> with binary search to lookup accounts and an ordered std::map<int, Account> which uses balanced binary search trees to get O(log n) runtimes for account searches.

Using this method of sorting is also helpful to list accounts by specific information such as account number or user name. By using MergeSort we can run in O(n log n) time and program transfers so that accounts are updating together and effectively so that no errors arise when the user is interacting with the program. By testing the program with multiple, different accounts we can compare runtime of searches and information retrieval with a small vs large number of accounts. This will demonstrate how the data structure and algorithm can help or hinder system performances. In this project, we will create an efficient banking system that lets users access their personal accounts and perform daily transactions, by testing different algorithms and data structures to see how they affect efficiency and reliability.