Project 3 Test Design Document: Craig Li (Section 8) Prerak Patel (Section 9)			
Test Case #	**Purpose of the Test Case**	**Input Data**	**Expected Output**
1	Test Case 1 tests the "Add Employee" button's functionality of our GUI for adding new employees to the database, starting with an empty database. The text area will display "Employee Added" upon successful addition of the employee to the database, and "Employee Not added" upon failure. This test case will attempt to add different employees to the database with differing inputs for each data field, including some that are left blank intentionally Case 1: Employee is successfully added to database Case 2: Employee is not added to database	Case 1, Instance 1: Name: John Smith Department: CS Date Hired: 3/10/2021 Position: Fulltime Salary: 65000 Case 2, Instance 1: Name: John Smith Department: CS Date Hired: 3/10/2021 Position: Fulltime Salary: 65000 Case 2, Instance 2: Name: John Smith Department: CS Date Hired: 3/10/2021 Position: Management (Director) Salary: 65000	Case 1, Instance 1: Message displays "Employee Added." Case 2, Instance 1: Message displays "Employee not added" Case 2, Instance 2: Message displays: "Employee Already Exists."
2	Test Case 2 tests the "Remove Employee" button's functionality, starting with a database with the employee from Case 1, Instance 1(John Smith) as the only entry. "Employee Removed" will be displayed in the text area upon successful removal, with "Employee database is empty." or "Employee does not exist." displayed for their respective situations. Test Cases are performed in the order of: 1. Case 2, Instance 1 2. Case 1, Instance 1 3. Case 2, Instance 2 Case 1: Employee is successfully removed from database Case 2: Employee is not removed from database (either employee does not exist, or the database is emtpy)	Case 1, Instance 1: Name: John Smith Department: CS Date Hired: 3/10/2021 Position: Fulltime Salary: 65000 Case 2, Instance 1: Name: Elliot Alderson Department: CS Date Hired: 3/10/2021 Position: Fulltime Salary: 65000 Case 2, Instance 2: Name: John Smith Department: CS Date Hired: 3/10/2021 Position: Fulltime Salary: 65000	Case 1, Instance 1: Message displays "Employee Removed." Case 2, Instance 1: Message displays "Employee doesn't Exist" Case 2, Instance 2: Message displays: "Employee database is empty."
3	Test Case 3 tests the "Set Hours" button, starting with a database of One Part time Employee matching Case 1, Instance 1. Upon successful setting of hours, the message will display "Hours Set.". If the hours are not able to be set due to invalid input, the message will display "Hours Not Set.". There are four test cases that use boundary testing to test the valid range of 0-100 hours for a Parttime employee. Case 1: Employee hours successfully set Case 2: Employee hours not set.	Case 1, Instance 1: Name: Tyrell Wellick Department: IT Date Hired: 3/10/2021 Position: Parttime Hours Worked: 0 Hourly Rate: 55.00 Case 1, Instance 2: Name: Tyrell Wellick Department: IT Date Hired: 3/10/2021 Position: Parttime Hours Worked: 100 Hourly Rate: 55.00 Case 2, Instance 1: Name: Tyrell Wellick Department: IT Date Hired: 3/10/2021 Position: Parttime Hours Worked: -1 Hourly Rate: 55.00 Case 2, Instance 2: Name: Tyrell Wellick Department: IT Date Hired: 3/10/2021 Position: Parttime Hours Worked: 101 Hourly Rate: 55.00	Case 1, Instance 1: Message displays "Working hours set." Case 1, Instance 2: Message displays "Working hours set." Case 2, Instance 1: Message displays "Working hours cannot be negative" Case 2, Instance 2: Message displays "Invalid Hours: over 100."
4	Test Case 4 test the "Process Payments" button, calculating payment rates per biweekly pay period for management, fulltime, and parttime employees. Parttime employees should have compensation equal to their hours worked multiplied by their hours worked during the pay period. Fulltime employees should have compensation equal to their salary divided by the pay periods per year. Management has compensation equal to their salary plus their management compensation divided by the pay periods per year. Several employees will be added to an empty database, and then their payments will be calculated and displayed via the print function. Case 1: Parttime Compensation Case 2: Fulltime Compensation Case 3: Management Compensation	Case 1, Instance 1: Name: Angela Moss Department: CS Date Hired: 3/10/2021 Position: Parttime Hours Worked: 80 Hourly Rate: 50.00 Case 1, Instance 2 (overttime): Name: Irving Department: IT Date Hired: 3/10/2021 Position: Parttime Hours Worked: 90 Hourly Rate: 50.00 Case 2, Instance 1: Name: Dominique DiPierro Department: ECE Date Hired: 3/10/2021 Position: Fulltime Salary: 75000 Case 3, Instance 1: Name: Philip Price Department: CS Date Hired: 3/10/2021 Position: Management (Director) Salary: 145000	Message displays: "Payment for employees processed" Printing the database displays: Angela Moss::CS::3/10/2021::Payment $4,000.00::PART TIME::Hourly Rate $50.00::Hours worked this period: 80 Irving::IT::3/10/2021::Payment $4,750.00::PART TIME::Hourly Rate $50.00::Hours worked this period: 90 Dominique Dipierro::ECE::3/10/2021::Payment $2,884.62::FULL TIME::Annual Salary $75,000.00 Philip Price::CS::3/10/2021::Payment $6,038.46::FULL TIME::Annual Salary $145,000.00::Director Compensation $461.54
5	Test Case 5 tests the capabilities of the File Import/Export Feature. Upon successful import, the information in the database imported should populate the database in the GUI. Upon successful export, the database will be exported to a file, without effecting the database contents in the GUI. Case 1: Successful I/O Operation Case 2: Unsuccessful I/O Operation	Case 1, Instance 1: Import database.txt provided Case 1, Instance 2: Export database loaded from database.txt to a new file. Case 2, Instance 1: Database is empty Case 2, Instance 2: User does not select a file to import	Case 1, Instance 1: database.txt is copied to GUI database Case 1, Instance 2: GUI database is written to a new text file Case 2, Instance 1: Message Displays: "Database is Empty" Case 2, Instance 2: Message Displays: "File Not Found"

6	Test Case 6 explores blank inputs. Cases will demonstrate that our program has been built with these invalid inputs in mind, and is able to handle all of them without our program crashing or exiting prematurely. Assume that operation that is not attempting to add an employee has a corresponding employee in the database already. Case 1: Blank Inputs	Case 1, Instance 1 (Add): Name: Department: CS Date Hired: 3/10/2021 Position: Fulltime Salary: 50000 Case 1, Instance 2 (Add): Name: Darlene Alderson Department: CS Date Hired: Position: Fulltime Salary: 50000 Case 1, Instance 3 (Add): Name: Darlene Alderson Department: CS Date Hired: 3/10/2021 Position: Fulltime Salary: Case 1, Instance 4 (SetHours): Name: Elliot Alderson Department: CS Date Hired: 3/10/2021 Position: Parttime Hours Worked: Hourly Rate: 80	Case 1, Instance 1: Message Displays: "Enter Name." "Employee Not Added." Case 1, Instance 2: Message Displays: "Invalid Date." "Employee not Added." Case 1, Instance 3: Message Displays: "Enter Annual Salary." "Employee Not Added." Case 1, Instance 4: Message Displays: "Enter Hours Worked"
7	Test Case 7 explores invalid inputs. Cases will demonstrate that our program has been built with these invalid inputs in mind, and is able to handle all of them without our program crashing or exiting prematurely. Case 1: Invalid Inputs	Case 1, Instance 1 (Add): Name: Cyril Figgis Department: CS Date Hired: 3/10/2021 Position: Fulltime Salary: -1 Case 1, Instance 2 (Add): Name: Mallory Archer Department: CS Date Hired: 12/31/1889 Position: Fulltime Salary: 50000 Case 1, Instance 3 (Add): Name: Mallory Archer Department: CS Date Hired: 3/10/2022 Position: Fulltime Salary: 50000	Case 1, Instance 1: Message Displays: "Salary Cannot be negative" "Employee not Added." Case 1, Instance 2: Message Displays: "Invalid Date." Case 1, Instance 3: Message Displays: "2022-3-10 is not a valid date!" "Employee not Added."
8	Test Case 8 explores bad inputs that throw exceptions. Cases will demonstrate that our program has been built with these invalid inputs in mind, and is able to handle all of them without our program crashing or exiting prematurely. Case 1: InputMisMatch Exception Case 2: ClassCastException Case 3: File Not Found Exception	Case 1, Instance 1 (Add): Name: Cheryl Tunt Department: CS Date Hired: 3/10/2021 Position: Fulltime Salary: Outlaw Country Case 1, Instance 2 (Add): Name: Sterling Archer Department: ECE Date Hired: 3/10/2021 Position: Fulltime Salary: 50000.75 Case 1, Instance 3 (setHours): Name: Algernop Krieger Department: ECE Date Hired: 3/10/2021 Position: Parttime Hours Worked: 97.45 Hourly Rate: 67.00 Case 2, Instance 1 (setHours): Name: Algernop Krieger Department: ECE Date Hired: 3/10/2021 Position: Fulltime Salary: 80000 Hours Worked: 40 Case 3, Instance1: Close out of fileChooser before choosing a file	Case 1, Instance 1: Message Displays: "Invalid Salary." "Employee Not Added" Case 1, Instance 2: Message Displays: "Invalid Salary." "Employee Not Added" Case 1, Instance 3: Message Displays: "Invalid Hours Worked." Case 2, Instance 1: Message Displays: "'Algernop Krieger' is not a Parttime Employee" Case 3, Instance 1: Message Displays: "File Not Found."