

### Software Methodologies Project 1 Test Document

Test Case #	Purpose of the Test Case	Input Data	Expected Output
1	Testing add button to see if empty input in any of the required fields will print an error message.	Name = “ ” Department = “CS” Date = “3/5/2018” Employment Type = “Parttime” Rate = 3	String appended to the text area stating that one of the required inputs is empty. In this case name.
2	Testing add button to see if it will print an error message if user inputs wrong data type for hourly rate.	Name = “Chris ” Department = “CS” Date = “3/5/2018” Employment Type = “Parttime” Rate = “String”	String appended to the text area telling the user that one of the required inputs is invalid and is not of type double.
3	Testing add button to see if it will print an error message if the user inputs the wrong data type for annual salary.	Name = “Toshan ” Department = “CS” Date = “3/5/2018” Employment Type = “Fulltime” Annual Salary = “String”	String appended to the text area telling the user that one of the required inputs is invalid and is not of type double.
4	Testing add button to see if it will print an message if the user inputs an invalid date.	Name = “Toshan ” Department = “CS” Date = “3/5/2022” Employment Type = “Fulltime” Annual Salary = 1000	String appended to the text area telling the user that the date is invalid and to try again.
5	Testing add button to see if it will add a fulltime employee when given the correct input.	Name = “Toshan ” Department = “CS” Date = “3/1/2021” Employment Type = “Fulltime” Annual Salary = 1000	String is appended to the text area on the GUI stating that the fulltime employee has been added.
6	Testing add button to see if it will add a management employee when given the	Name = “Toshan ” Department = “CS” Date = “3/1/2021”	String is appended to the text area on the GUI stating that the

	correct input.	Employment Type = "Management" Annual Salary = 1000 Management Role = "Manager"	management employee has been added.
7	Testing add button to see if it will add a parttime employee when given the correct input.	Name = "Toshan " Department = "CS" Date = "3/1/2021" Employment Type = "Parttime" Rate = 1000	String is appended to the text area on the GUI stating that the parttime employee has been added.
8	Testing add button to see if it will add an employee when the employee has already been added.	Input1 Name = "Toshan " Department = "CS" Date = "3/1/2021" Employment Type = "Fulltime" Annual Salary = 1000  Input 2 Name = "Toshan " Department = "CS" Date = "3/1/2021" Employment Type = "Fulltime" Annual Salary = 1000	String is appended to the text area on the GUI stating that the employee is already in the list.
9	Testing add button to see if it will handle the error for a negative salary or pay rate.	Name = "Toshan " Department = "CS" Date = "3/1/2021" Employment Type = "Fulltime" Annual Salary = -100	String is appended to the text area telling the user that the salary cannot be negative.
10	Testing remove button to see if an empty input in any of the required fields will print an error message.	Name = "Chris" Department = " " Date = "3/5/2018"	String is appended to the text area telling the user that one of the required fields is empty. In this case department.
11	Testing remove button to see if it will print an error message if the user inputs an invalid date.	Name = "Chris" Department = "CS " Date = "3/5/2023"	String is appended to the text area telling the user that the date is invalid and to try

			again.
12	Testing remove button to see if it will handle the case when the employee database is empty	Name = "Chris" Department = "CS " Date = "4/5/2018"	String is appended to the text area on the GUI stating that the employee database is empty.
13	Testing remove button to see if it will handle the case when the employee does not exist in the database.	Input1 ADD EMPLOYEE Name = "Toshan " Department = "CS" Date = "3/1/2021" Employment Type = "Fulltime" Annual Salary = 1000  Input 2 REMOVE EMPLOYEE Name = "Chris" Department = "CS" Date = "3/1/2018"	String is appended to the text area on the GUI stating that the employee does not exist.
14	Testing remove button to see if it will remove a given fulltime employee that exists in the database	Input1 ADD EMPLOYEE Name = "Toshan " Department = "CS" Date = "3/1/2021" Employment Type = "Fulltime" Annual Salary = 1000  Input 2 REMOVE EMPLOYEE Name = "Toshan" Department = "CS" Date = "3/1/2021"	String is appended to the text area on the GUI stating that the employee has been removed
15	Testing remove button to see if it will remove a given management employee that exists in the database	Input1 ADD EMPLOYEE Name = "Toshan " Department = "CS" Date = "3/1/2021" Employment Type =	String is appended to the text area on the GUI stating that the employee has been removed

		"Management" Annual Salary = 1000 Management Role = "Manager" Input2 REMOVE EMPLOYEE Name = "Toshan" Department = "CS" Date = "3/1/2021"	
16	Testing remove button to see if it will remove a given parttime employee that exists in the database	Input 1 ADD EMPLOYEE Name = "Toshan " Department = "CS" Date = "3/1/2021" Employment Type = "Parttime" Rate = 1000 Input 2 REMOVE EMPLOYEE Name = "Toshan" Department = "CS" Date = "3/1/2021"	String is appended to the text area on the GUI stating that the employee has been removed
17	Testing the set hours button to see if empty input in any of the required fields will print an error message.	Name = "Chris" Department = "CS" Employment type = " Date = "3/5/2018" Hours worked = 23	String is appended to the text area telling the user that one of the required fields is empty. In this case employment type.
18	Testing set hours button to see if it will print an error message if the user inputs an invalid date.	Name = "Chris" Department = "CS " Employment type = "Parttime" Date = "3/5/2023" Hours worked = 44	String is appended to the text area telling the user that the date is invalid and to try again.
19	Testing set hours button to see if it will print an error message if the employment type is anything else besides parttime.	Name = "Chris" Department = "CS " Employment type = "Fulltime" Date = "3/5/2020" Hours worked = 44	String is appended to the text area telling the user that it cannot set hours for Management or Fulltime only

			Parttime
20	Testing set hours button to see if it will print an error message if the input for hours worked is not a valid type int.	Name = "Chris" Department = "CS " Employment type = "Parttime" Date = "3/5/2020" Hours worked = "String"	String is appended to the text area telling the user that hours worked is not a valid type int.
21	Testing set hours button to see how it will handle if the employee database is empty.	Name = "Chris" Department = "CS " Employment type = "Parttime" Date = "3/5/2020" Hours worked = "23"	String is appended to the text area on the GUI stating that the employee database is empty.
22	Testing set hours button to see if it will set hours for an employee that does not exist.	Input1 ADD EMPLOYEE Name = "Toshan " Department = "CS" Date = "3/1/2021" Employment Type = "Fulltime" Annual Salary = 1000  Input 2 SET HOURS Name = "Chris" Department = "CS " Employment type = "Parttime" Date = "3/5/2020" Hours worked = "23"	String is appended to the text area on the GUI stating that the employee does not exist.
23	Testing set hours button to see if it will print an error message if the hours inputted is negative	Name = "Chris" Department = "CS " Employment type = "Parttime" Date = "3/5/2020" Hours worked = "-23"	String is appended to the text area on the GUI stating that the hours worked cannot be negative
24	Testing set hours button to see if it will print an message if the hours inputted is greater than	Name = "Chris" Department = "CS " Employment type =	String is appended to the text area on the GUI stating that the

	100	“Parttime” Date = “3/5/2020” Hours worked = “105”	hours worked cannot be over 100.
25	Testing set hours button to see if it will set hours for an employee appropriately	Input1 ADD EMPLOYEE Name = “Chris ” Department = “CS” Date = “3/1/2021” Employment Type = “Parttime” Hourly Rate = 1000  Input 2 Name = “Chris” Department = “CS “ Employment type = “Parttime” Date = “3/1/2021” Hours worked = “23”	String is appended to the text area on the GUI stating that the working hours has been set.
26	Testing calculate payments button when employee database is empty	No Input needed just press calculate payment	String is appended to the text area on the GUI stating that the employee database is empty.
27	Testing calculate payments button when employee database is not empty	Input1 ADD EMPLOYEE Name = “Chris ” Department = “CS” Date = “3/1/2021” Employment Type = “Parttime” Hourly Rate = 3  Input 2 Name = “Chris” Department = “CS “ Employment type = “Parttime” Date = “3/1/2021” Hours worked = “23”	String is appended to the text area on the GUI stating that the calculating payments for employees are done. There is a correct value of \$69 for payment.

		Input 3 Action event click calculate payment	
28	Testing Import Button with empty txt	Pressing Import button and choosing an empty txt file from file chooser	It will not add anything to the database and the app will continue running.
29	Testing Import Button with non txt	Pressing Import and attempting to select a non txt file from file chooser	File chooser does not allow you to select a non txt file.
30	Testing Import Button with database.txt	Pressing Import and selecting database.txt from file chooser	Properly adds all employees and properly appends string to text area for each added employee
31	Testing Export Button with empty database	Pressing the export button when the database is empty.	String is appended to the text area stating that the database cannot be exported because it is empty.
32	Testing Export Button to a new file	Pressing the export button when writing to a new file.	New file is created and the database is properly exported to the new file. String appended to the text area stating that the database was properly exported to the designated file.
33	Testing Export Button to a existing file	Pressing the export button when writing to an existing file	Alert pops up asking if you would like to replace the existing file. Handles appropriately based on decision. If the user wants to replace, the existing file is overwritten with the

			<p>new database. If the user does not want to replace, they are asked to reenter the target file. String appended to the text area stating that the database was properly exported to the designated file.</p>
--	--	--	--