		le	Let 1						
Function Name	Test ID	Description	Status	Test Data 1	Test Data 2	Test Data 3	Test Data 4	Test Data 5	Expected output
ogin		These Tests Test the login function and checks if it returns a user as an output		Username	Password				
	customerlogin01	The username is right and password is wrong, should return null	PASS	admin	admin				NULL
	customerlogin02	The username is wrong and password is right, should return null	PASS	admin1	password				NULL
	customerlogin03	The username is wrong and password is wrong, should return null	PASS	admin1	admin1				NULL
			PASS						
	customerlogin04	The username is right and password is right, should return the user	PASS	admin	password				USER
ogin		Checks if if an owner can login		Username	Password				
	ownerlogin01	The username is right and password is wrong, should return null	PASS	Ownertest	password				NULL
	ownerlogin02	The username is wrong and password is right, should return null	PASS	Ownertest1	1234				NULL
	ownerlogin03	The username is wrong and password is wrong, should return null	PASS	Ownertest1	password				NULL
			PASS		1234				OWNER
	ownerlogin04	The username is right and password is right, should return the user	PASS	Ownertest	1234				OWNER
gister		Checks if bad input will pass through validation checks		Username	Password	Name	Address	Phone Number	
	customerregister01	All details are vaild but the username already exists in the database, returns null	PASS	Ownertest	password	Russell	12 Melbourne	03987656789	NULL
	customerregister02	All details are vaild, creates the user in the database	PASS	admin	password	Russell	12 Melbourne	03987656789	USER
		All details are valid but the name, returns null	PASS	admin		Russell1234	12 Melbourne	03987656789	NULL
	customerregister03				password				
	customerregister04	All details are valid but the address, returns null	PASS	admin	password	Russell	Fake888***	03987656789	NULL
	customerregister05	All details are valid but the phone number, returns null	PASS	admin	password	Russell	12 Melbourne	abcd	NULL
dEmployee		Checks if bad input will pass through validation checks		Name	Phone Number	Address			
	addemployeetest01	All details are valid but the name, returns false	PASS	Fred1234	03123456789	12 Melbourne			FALSE
	addemployeetest02	All details are valid but the phone number, returns false	PASS	Fred	asdf	12 Melbourne			FALSE
	addemployeetest03	All details are valid but the address, returns false	PASS	Fred	03123456789	Melbourne			FALSE
	addemployeetest04	All details are vaild, creates the employee in the database, returns true	PASS	Fred	03123456789	12 Melbourne			TRUE
tEmployeeBookingAvailability									
,,gr.vanability	getEmployeeBookingAvailability04	Get employee booking availability for employee 0 after 0.	PASS						
		Get employee booking availability for employee 3 after 0.	PASS						
	getEmployeeBookingAvailability03	Get employee booking availability for employee 3 after 2147410000.	PASS						
tWorkingTimes									
gg	antiMarkinaTimenO1	ant warking times for appleves 1	PASS						
	getWorkingTimes01	get working times for employee 1							
	getWorkingTimes02	get working times for employee 2	PASS						
	getWorkingTimes03	get working times for employee 3	PASS						
	getWorkingTimes04	get working times for non-existent employee '-1', should return null	PASS						
	getWorkingTimes05	get working times for non-existent employee '10', should return null	PASS						
	3	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,							
dalE-sealers -									
ddEmployee									
	addEmployeeTest01	Add employee with no owner	PASS						
	addEmployeeTest02	Add employee with invalid phone number	PASS						
	addEmployeeTest03	Add employee with no username	PASS						
	addEmployeeTest04	Add employee with invalid username	PASS						
		1 1	PASS						
	addEmployeeTest05	Add employee with no address							
	addEmployeeTest06	Add valid employee	PASS						
ldWorkingTime									
· · · · · · · · · · · · · · · · · · ·	addWorkingTime01	Add a working time where the end time is before the start time	PASS						
		-	PASS						
	addWorkingTime02	Add a working time of duration 0							
	addWorkingTime03	Add a working time with a negative start time	PASS						
	addWorkingTime04	Add a valid working time	PASS						
noveBooking									
	removeBookings01	Remove a non-existent booking	PASS						
		-							
	removeBookings02	Remove a booking without a business	PASS						
	removeBookings03	Remove a booking that does exist	PASS						
	removeBookings04	Remove a booking with an invalid id	PASS						
	removeBookings05	Remove a booking with an valid id	PASS						
10									
dService									
	addService01	Add service with an invalid name	PASS						
	addService02	Add a service that already exists	PASS						
	addService03	Add a service with a negative duration	PASS						
			PASS						
	addService04	Add a service with a negative cost							
	addService05	Add a valid service	PASS						
	addService06	Add a service with a duration that isn't dividable by 30	PASS						
dNewBooking									

Function Name	Test ID	Description	Status	Test Data 1	Test Data 2	Test Data 3	Test Data 4	Test Data 5	Expected output	
	addNewBooking02	Add a booking that already exists	PASS							
	addNewBooking03	Add a booking that overlaps with another booking for the same employee	PASS							
	addNewBooking04	Add a valid booking	PASS							
	addNewBooking05	Add a booking for a non-existent user	PASS							
	addNewBooking06	Add a booking where there is no slot available	PASS							
etEmployeeList										
	getEmployeeList01	Employee list returns the right employees	PASS							
getSummaryOfBookings										
	getSummaryOfBookings01	Correct bookings returned	PASS							
etCurrentBookings										
	getCurrentBookings01	Gets all bookings after system time	PASS							
	getCurrentBookings02	Gets all bookings after system time after adding a new booking after system time	PASS							
etCustomerList										
	getCustomerList01	Returns correct customers	PASS							
getServicesList										
	getServicesList01	Returns correct services	PASS							

	1	I=	I=							
Function Name	Test ID	Description	Status	Test Data 1	Test Data 2	Test Data 3	Test Data 4	Test Data 5	Expected output	
searchUser				Username						
	searchUser1	Check if the user does not exist in the database will return null	PASS	Ownertest123					null	
	searchUser2	Check if Owner exists in the database and login	PASS	Ownertest					OWNER	
	searchUser3	Check if correct user is being returned	PASS							
	searchUser4	Check if correct user is being returned	PASS							
	searchUser5	Check that the user does not exist	PASS							
	searchUser6	Check that the user does not exist	PASS							
	Scarcifoscio	Official trial tric docs not exist	1700							
addEmployee										
	addEmployee1	employee added doesn't share timetable with another employee	PASS							
	addEmployee2	valid employee	PASS							
	addEmployee3	add employee with duplicate id	PASS							
	addEmployee4	add employee with a non existent business	PASS							
	addEmployee5	valid employee	PASS							
getAllServices										
getAllServices		check length of list	DAGO							
	getAllServices1	ů .	PASS							
	getAllServices2	check that all the services exist in the list	PASS							
addNewBooking										
	addNewBooking1	Add a valid booking	PASS							
	addNewBooking2	add a booking in the same slot as the last booking	PASS							
	addNewBooking3	Add a booking in an invalid slot	PASS							
	addNewBooking4	Add a booking right after another booking ends	PASS							
	addNewBooking5	Add a booking with a non-existent customer	PASS							
		Add a booking with a non-existent customer Add a booking with a non-existent employee	PASS							
	addNewBooking6									
	addNewBooking7	Add a booking with a non-existent service	PASS							
getAvailableTimes										
	getAvailableTimes1	get available times for employee of id 0	PASS							
	getAvailableTimes2	get available times for employee of id 1	PASS							
	getAvailableTimes3	get available times for employee of id 2	PASS							
	getAvailableTimes4	get available times for employee of id 3	PASS							
	getAvailableTimes5	get available times for employee of id 4	PASS							
	getAvailableTimes6	get available times for employee of id 5	PASS							
	getAvaliable Fifteso	get available times for employee of to 5	FAGO							
addShift										
	addShift1	Add shift where end time is before the start time	PASS							
	addShift2	Add shift of duration 0, should not pass	PASS							
	addShift3	Add shift with a negative start time (Allows start time to go into previous day	PASS							
	addShift4	Add a valid shift	PASS							
removeBooking										
removeBooking	removeBooking1	remove booking with negative id	PASS							
	removeBooking2	remove booking without specifying business	PASS							
	removeBooking3	remove booking with valid details	PASS							
	removeBooking4	remove booking with non existent id	PASS							
	removeBooking5	remove booking with id 0	PASS							
getEmployeeBookingAvailability										
	getEmployeeBookingAvailability1	Get all booking availability slots which are available for employee 0	PASS							
	getEmployeeBookingAvailability2	Get all booking availability slots which are available for employee 3	PASS							
	getEmployeeBookingAvailability3	Get all bookings on or after 2147410000 for employee 3, which should be none	PASS							
			PASS							
	getEmployeeBookingAvailability4	Get all bookings on or after 1497522600 for employee 3, which should be 1								
	getEmployeeBookingAvailability5	Only booking available after 1497522600 for employee 3 should start at 1497508200	PASS							
	getEmployeeBookingAvailability6	Get all bookings on or after 1497522601 for employee 3, which should be none	PASS							
getShift										
	getShift1	get shifts of employee of id 0	PASS							
	getShift2	get shifts of employee of id 1	PASS							
	getShift3	get shifts of employee of id 2	PASS							
	g5	garania a ampioyoo or o a								
gotEmployagi ist										
getEmployeeList	AF	Observation to the line and the line	PAGG							
	getEmployeeList1	Check the length of the list	PASS							
	getEmployeeList2	Check that all employees exist in order	PASS							
	getEmployeeList3	Check that the list exists	PASS							
getAllCustomers										
	getAllCustomers1	Check the length of the list	PASS							
	3-1 2 00 (01110101		1.1.1.1.1							

Test ID	Description	Status	Test Dat	1 Test Data 2	Test Data 3	Test Data 4	Test Data 5	Expected output
getAllCustomers2	Check that all customers exist in order	PASS						
getAllCustomers3	Check that the list exists	PASS						
getBookingsAfter1	Get bookings after 0, there should be 8	PASS						
getBookingsAfter2	Get bookings after 1497506400, there should be 8	PASS						
getBookingsAfter3	Get bookings after 1497558600, there should be at least 1	PASS						
getBookingsAfter4	Get bookings after 1497558601, should return null	PASS						
getBookingsAfter5	Get bookings after 1497558599, there should be at least 1	PASS						
getBookingsAfter6	Get bookings after 1497506401, there should be 7	PASS						
getEmployeeAvailability1	Make sure that available bookings match expected for employee 0	PASS						
getEmployeeAvailability2	Make sure that available bookings match expected for employee 1	PASS						
getEmployeeAvailability3	Make sure that available bookings match expected for employee 2	PASS						
addCustomer1	Add valid customer	PASS						
addCustomer2	Add customer with invalid username	PASS						
addCustomer3	Add customer with invlaid phone number	PASS						
addCustomer4	Add valid customer	PASS						
addCustomer5	Add customer without a password	PASS						
addCustomer6	Add customer without a username	PASS						
authenticate1	Authenticate with correct details	PASS						
authenticate2	Authenticate with correct details	PASS						
authenticate3	Authenticate with correct details	PASS						
authenticate4	Authenticate with incorrect username	PASS						
authenticate5	Authenticate with incorrect password	PASS						
authenticate6	Authenticate without any details	PASS						
	getBookingsAfter1 getBookingsAfter1 getBookingsAfter2 getBookingsAfter2 getBookingsAfter3 getBookingsAfter3 getBookingsAfter4 getBookingsAfter6 getEmployeeAvailability1 getEmployeeAvailability2 getEmployeeAvailability3 addCustomer1 addCustomer2 addCustomer3 addCustomer4 addCustomer6 authenticate1 authenticate2 authenticate2 authenticate4 authenticate4	getBookingsAfter1 getBookingsAfter2 Get bookings after 0, there should be 8 getBookingsAfter2 Get bookings after 1497506400, there should be 8 getBookingsAfter3 Get bookings after 1497506600, there should be at least 1 getBookingsAfter4 Get bookings after 1497558601, should return null getBookingsAfter5 Get bookings after 1497558691, should return null getBookingsAfter6 Get bookings after 1497558699, there should be at least 1 getBookingsAfter6 Get bookings after 1497506401, there should be 7 getEmployeeAvailability1 Make sure that available bookings match expected for employee 0 getEmployeeAvailability2 Make sure that available bookings match expected for employee 1 getEmployeeAvailability3 Make sure that available bookings match expected for employee 2 addCustomer1 Add valid customer addCustomer2 Add customer with invalid username addCustomer3 Add customer with invalid phone number addCustomer4 Add valid customer addCustomer5 Add customer without a password addCustomer6 Add customer without a username authenticate1 authenticate1 authenticate2 Authenticate with correct details authenticate3 Authenticate with correct details authenticate4 Authenticate with incorrect username authenticate5 Authenticate with incorrect username authenticate5 Authenticate with incorrect username	getBookingsAfter1 Get bookings after 0, there should be 8 getBookingsAfter2 Get bookings after 1497506400, there should be 8 getBookingsAfter3 Get bookings after 1497506400, there should be 8 getBookingsAfter3 Get bookings after 1497558600, there should be at least 1 pASS getBookingsAfter4 Get bookings after 1497558600, there should be at least 1 pASS getBookingsAfter5 Get bookings after 1497558600, there should be at least 1 pASS getBookingsAfter6 Get bookings after 1497568600, there should be at least 1 pASS getBookingsAfter6 Get bookings after 1497506401, there should be 7 pASS getEmployeeAvailability1 Make sure that available bookings match expected for employee 0 pASS getEmployeeAvailability2 Make sure that available bookings match expected for employee 1 pASS getEmployeeAvailability3 Make sure that available bookings match expected for employee 2 pASS addCustomer1 Add valid customer addCustomer2 Add customer with invalid username pASS addCustomer3 Add customer with invalid username pASS addCustomer4 Add valid customer Add valid customer with invalid phone number pASS addCustomer5 Add customer with invalid phone number pASS addCustomer6 Add customer without a password pASS addCustomer6 Add customer without a username pASS addCustomer6 Add customer without a username pASS authenticate1 Authenticate with correct details pASS authenticate3 Authenticate with incorrect password pASS authenticate6 Authenticate with incorrect password pASS authenticate6 Authenticate with incorrect password pASS	getBookingsAfter1 Get bookings after 0, there should be 8 getBookingsAfter2 Get bookings after 1497508400, there should be 8 getBookingsAfter3 Get bookings after 1497508400, there should be 8 getBookingsAfter3 Get bookings after 1497508600, there should be at least 1 pASS getBookingsAfter4 Get bookings after 1497508601, should return null pASS getBookingsAfter5 Get bookings after 1497508601, should return null pASS getBookingsAfter6 Get bookings after 1497508601, should return null pASS getBookingsAfter6 Get bookings after 1497508601, there should be 7 pASS getBookingsAfter6 Get bookings after 1497508401, there should be 7 pASS getEmployeeAvailability1 Make sure that available bookings match expected for employee 0 getEmployeeAvailability2 Make sure that available bookings match expected for employee 1 getEmployeeAvailability3 Make sure that available bookings match expected for employee 2 getEmployeeAvailability3 Make sure that available bookings match expected for employee 2 getEmployeeAvailability3 Make sure that available bookings match expected for employee 2 getEmployeeAvailability3 Make sure that available bookings match expected for employee 2 pASS getContinued PASS addCustomer1 Add vaild customer Add vaild customer PASS addCustomer2 Add customer with invalid username PASS addCustomer3 Add customer with invalid phone number pASS addCustomer6 Add valid customer Add valid customer with invalid phone number pASS addCustomer6 Add customer without a username PASS authenticate1 Authenticate with correct details pASS authenticate2 Authenticate with correct details pASS authenticate3 Authenticate with incorrect username PASS authenticate4 Authenticate with incorrect password PASS authenticate5 Authenticate with incorrect password PASS	getBookingsAfter1 Get bookings after 0, there should be 8 PASS getBookingsAfter2 Get bookings after 1497506400, there should be 8 PASS getBookingsAfter3 Get bookings after 1497556400, there should be 8 PASS getBookingsAfter3 Get bookings after 1497556800, there should be at least 1 PASS getBookingsAfter4 Get bookings after 1497556800, there should be at least 1 PASS getBookingsAfter5 Get bookings after 1497556800, there should be at least 1 PASS getBookingsAfter6 Get bookings after 1497556800, there should be at least 1 PASS getBookingsAfter6 Get bookings after 149756800, there should be 7 PASS getBookingsAfter6 Get bookings after 1497560401, there should be 7 PASS getEmployeeAvailability1 Make sure that available bookings match expected for employee 0 PASS getEmployeeAvailability2 Make sure that available bookings match expected for employee 1 PASS getEmployeeAvailability3 Make sure that available bookings match expected for employee 2 PASS addCustomer1 Add valid customer PASS addCustomer2 Add customer with invalid usemame PASS addCustomer3 Add customer with invalid usemame PASS addCustomer3 Add customer with invalid phone number PASS addCustomer6 Add customer without a usemame PASS authenticate1 Authenticate with correct details PASS authenticate2 Authenticate with correct details PASS authenticate3 Authenticate with correct details PASS authenticate6 Authenticate with correct usemame PASS	getBookingsAfter1 Get bookings after 0, there should be 8 PASS getBookingsAfter2 Get bookings after 1497506400, there should be 8 PASS getBookingsAfter3 Get bookings after 1497506400, there should be 8 PASS getBookingsAfter3 Get bookings after 1497506400, there should be at least 1 PASS getBookingsAfter4 Get bookings after 14975506800, there should be at least 1 PASS getBookingsAfter5 Get bookings after 14975506401, there should be at least 1 PASS getBookingsAfter6 Get bookings after 14975506401, there should be at least 1 PASS getBookingsAfter6 Get bookings after 14975506401, there should be 7 PASS getBookingsAfter6 Get bookings after 1497506401, there should be 7 PASS getEmployeeAvailability1 Make sure that available bookings match expected for employee 0 PASS getEmployeeAvailability2 Make sure that available bookings match expected for employee 1 PASS getEmployeeAvailability3 Make sure that available bookings match expected for employee 2 PASS addCustomer1 Add valid customer PASS addCustomer2 Add customer with invalid username PASS addCustomer3 Add customer with invalid username PASS addCustomer6 Add customer with invalid username PASS addCustomer6 Add customer without at username PASS addCustomer6 Add customer without at username PASS authenticate1 Authenticate with correct details PASS authenticate2 Authenticate with correct details PASS authenticate3 Authenticate with correct details PASS authenticate6 Authenticate with correct details PASS	gelBookingsAfter1 Get bookings after 0, there should be 8 PASS gelBookingsAfter2 Get bookings after 1497506400, there should be 8 PASS gelBookingsAfter3 Get bookings after 1497506400, there should be 8 PASS gelBookingsAfter3 Get bookings after 1497506400, there should be 8 PASS gelBookingsAfter3 Get bookings after 149758001, should return rull PASS gelBookingsAfter6 Get bookings after 149758001, should return rull PASS gelBookingsAfter6 Get bookings after 149758099, there should be 17 PASS gelBookingsAfter6 Get bookings after 149758099, there should be 7 PASS gelBookingsAfter6 Get bookings after 1497506401, there should be 7 PASS gelBookingsAfter6 Get bookings after 1497506401, there should be 7 PASS gelEmptypeeAvailability1 Make sure that available bookings match expected for emptypee 0 PASS gelEmptypeeAvailability2 Make sure that available bookings match expected for emptypee 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptypee 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptypee 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptypee 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptypee 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptypee 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptypee 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptypee 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptypee 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptypee 2 PASS gelEmptypeeAvailability4 gelEmptypeeAvailability5 gelEmptypeeAvailability6 ge	gelBockingsAfter1 Get bookings after 0, there should be 8 PASS gelBockingsAfter2 Get bookings after 1497506400, there should be 8 PASS gelBockingsAfter3 Get bookings after 1497506400, there should be 1 eteat 1 PASS gelBockingsAfter3 Get bookings after 1497506400, there should be a teat 1 PASS gelBockingsAfter3 Get bookings after 1497506400, there should be a teat 1 PASS gelBockingsAfter3 Get bookings after 1497506401, there should be a teat 1 PASS gelBockingsAfter3 Get bookings after 1497506401, there should be 7 PASS gelBockingsAfter3 Get bookings after 1497506401, there should be 7 PASS gelBockingsAfter3 Get bookings after 1497506401, there should be 7 PASS gelEmptypeeAvailability1 Make sure that available bookings match expected for emptype 0 PASS gelEmptypeeAvailability2 Make sure that available bookings match expected for emptype 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptype 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptype 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptype 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptype 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptype 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptype 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptype 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptype 2 PASS gelEmptypeeAvailability3 Make sure that available bookings match expected for emptype 2 PASS gelEmptypeeAvailability4 Make sure that available bookings match expected for emptype 2 PASS gelEmptypeeAvailability4 Make sure that available bookings match expected for emptypee 2 PASS gelEmptypeeAvailability4 Make sure that available bookings match expected for emptypee 2 PASS gelEmptypeeAvailability4 Make sure that available bookings match expected

Function Name	Test ID	Description	Status	Test Data 1	Test Data 2	Test Data 3	Test Data 4	Test Data 5	Expected output
convertSecondsToDay()		Checks the conversion of se	conds to days	Seconds					
	testConvertSecondsToDay1	Checks if it converts to tuesday	PASS	86401					TRUE
	testConvertSecondsToDay2	Checks if it converts to monda	PASS	86399					TRUE
	testConvertSecondsToDay3	Checks if it converts to friday	PASS	345600					TRUE
	testConvertSecondsToDay4	Checks if it converts to tuesday	PASS	90000					TRUE
	testConvertSecondsToDay5	Checks if it converts to tuesday	PASS	100000					TRUE
	testConvertSecondsToDay6	Checks if it converts to Sunday	PASS	604799					TRUE
	testConvertSecondsToDay7	Checks if it converts to Monda	PASS	1					TRUE
convertDayToSeconds()				Day String					
	testConvertDayToSeconds1	Checks if it converts to 0	PASS	Monday					TRUE
	testConvertDayToSeconds2	Checks if it converts to 86400	PASS	Tuesday					TRUE
	testConvertDayToSeconds3	Checks if it converts to 172800	PASS	Wednesday					TRUE
	testConvertDayToSeconds4	Checks if it converts to 259200	PASS	Thursday					TRUE
	testConvertDayToSeconds5	Checks if it converts to 345600	PASS	Friday					TRUE
	testConvertDayToSeconds6	Checks if it converts to 518400	PASS	Sunday					TRUE
	testConvertDayToSeconds7	Checks if it converts to 0	PASS	Not a valid day					TRUE
hecklsValidWeekday()									
	checkValidDay1	Checks if "Monday" is a valid of	PASS						TRUE
	checkValidDay2	Checks if "Sunday" is a valid d	PASS						TRUE
	checkValidDay3	Checks if "WEDNESDAY" is a	PASS						TRUE
	checkValidDay4	Checks if "friday" is a valid dat	PASS						TRUE
	checkValidDay5	Checks if "MON" is a valid date	PASS						FALSE
	checkValidDay6	Checks if "Saturnday" is a valid	PASS						FALSE
get24HrTimeFromWeekTime()									
	testConvertTimeTo24hrTime1	Check if 60 seconds is equal to	PASS						TRUE
	testConvertTimeTo24hrTime2	Check if 3600 seconds is equa	PASS						TRUE
	testConvertTimeTo24hrTime3	Check if 86399 seconds is equ	PASS						TRUE
	testConvertTimeTo24hrTime4	Check if 86340 seconds is equ	PASS						TRUE

Test ID	Description	Status	Test Data 1	Test Data 2	Test Data 3	Test Data 4	Test Data 5	Expected output
	Checks to see if it merges 2 timetables together		Timetable 1	Timetable 2				
mergeTimetable1	Checks the merging of and empty timetable and filled timetable	PASS	*Empty*	90000,93600 97	200,100600 21600,43200			TRUE
mergeTimetable2	Checks the merging of 2 tables with contradicting times	PASS	93500,97300	90000,93600 97	200,100600 216	00,43200		TRUE
mergeTimetable3	Checks the merging of 2 tables with contradicting times	PASS	90000,93600 97	72 93500,97300 21	600,43200			TRUE
mergeTimetable4	Checks the merging of 2 tables with different times	PASS	21600,43200	90000,93600 97	200,100600			TRUE
mergeTimetable5	Checks the merging of 2 tables with different times, only one period each	PASS	90000,93600	21600,43200				TRUE
mergeTimetable6	Checks the merging of 2 tables with different times, only one period each	PASS	40000,43600	91600,123200				TRUE
				Period 2				
addPeriod1	Checks to see if the new period is added to an empty timetable	PASS	90000,93600					TRUE
addPeriod2	Checks to see if the new period is added to an empty timetable	PASS	40000,43600					TRUE
addPeriod3	Checks to see if the new period is added to a timetable with a period in it	PASS	90000,93600	40000,43600				TRUE
addPeriod4	Checks to see if the new period is added to a timetable with a period in it	PASS	40000,43600	90000,93600				TRUE
	Removes periods to a timetable		Period 1	Period 2	Time Table			
removePeriod1	Checks to see if the period is romoved from the timetable	PASS	90000,93600		90000,93600 4	0000,43600		TRUE
removePeriod2	Checks to see if the period is romoved from the timetable	PASS	40000,43600		90000,93600 4	0000,43600		TRUE
removePeriod3	Checks to see if the all periods are romoved from the timetable	PASS	90000,93600	40000,43600	90000,93600 4	0000,43600		TRUE
removePeriod4	Checks to see if it returns false if trying to remove a period that is not there	PASS	40000,43600	90000,93600				TRUE
	Checks the output of the function to ensure its in the right format		Timetable					
toStringArray	Runs the methods and iterates through the elemets of the array checking each of them	PASS	90000,93600 9	7200,100600 2160	0,43200			TRUE
	merge Timetable1 merge Timetable2 merge Timetable3 merge Timetable4 merge Timetable4 merge Timetable5 merge Timetable6 addPeriod1 addPeriod2 addPeriod2 addPeriod3 addPeriod4 removePeriod1 removePeriod2 removePeriod3 removePeriod3 removePeriod4	Checks to see if it merges 2 timetables together merge Timetable1 Checks the merging of and empty timetable and filled timetable merge Timetable2 Checks the merging of 2 tables with contradicting times merge Timetable3 Checks the merging of 2 tables with contradicting times merge Timetable4 Checks the merging of 2 tables with different times merge Timetable5 Checks the merging of 2 tables with different times, only one period each merge Timetable6 Checks the merging of 2 tables with different times, only one period each Adds periods to a timetable Adds periods to a timetable addPeriod1 Checks to see if the new period is added to an empty timetable addPeriod2 Checks to see if the new period is added to an empty timetable addPeriod3 Checks to see if the new period is added to a timetable with a period in it Removes periods to a timetable removePeriod1 Checks to see if the period is romoved from the timetable removePeriod2 Checks to see if the period is romoved from the timetable removePeriod3 Checks to see if the period is romoved from the timetable removePeriod3 Checks to see if the period is romoved from the timetable removePeriod3 Checks to see if the period is romoved from the timetable removePeriod4 Checks to see if the turns false if trying to remove a period that is not there Checks to see if it returns false if trying to remove a period that is not there	Checks to see if it merges 2 timetables together mergeTimetable1 Checks the merging of 2 tables with contradicting times PASS mergeTimetable3 Checks the merging of 2 tables with contradicting times PASS mergeTimetable4 Checks the merging of 2 tables with contradicting times PASS mergeTimetable5 Checks the merging of 2 tables with different times PASS mergeTimetable6 Checks the merging of 2 tables with different times, only one period each PASS mergeTimetable6 Checks the merging of 2 tables with different times, only one period each PASS mergeTimetable6 Checks to see if the new period is added to an empty timetable PASS Adds periods to a timetable addPeriod1 Checks to see if the new period is added to an empty timetable PASS addPeriod2 Checks to see if the new period is added to a timetable with a period in it PASS addPeriod4 Checks to see if the new period is added to a timetable with a period in it PASS Removes periods to a timetable RemovePeriod1 Checks to see if the period is romoved from the timetable PASS removePeriod2 Checks to see if the period is romoved from the timetable PASS removePeriod3 Checks to see if the period are romoved from the timetable PASS removePeriod4 Checks to see if the period are romoved from the timetable PASS removePeriod3 Checks to see if the period are romoved from the timetable PASS removePeriod4 Checks to see if the period are romoved from the timetable PASS removePeriod4 Checks to see if the period are romoved from the timetable PASS removePeriod4 Checks to see if the period are romoved from the timetable PASS	Checks to see if it merges 2 timetables together mergeTimetable1 Checks the merging of and empty timetable and filled timetable PASS PASS 93500,97300 mergeTimetable2 Checks the merging of 2 tables with contradicting times PASS 93500,97300 mergeTimetable3 Checks the merging of 2 tables with contradicting times PASS 93500,97300 mergeTimetable4 Checks the merging of 2 tables with outradicting times PASS 9450 mergeTimetable5 Checks the merging of 2 tables with different times PASS 9500,93600 mergeTimetable6 Checks the merging of 2 tables with different times, only one period each PASS 9500,93600 mergeTimetable6 Checks the merging of 2 tables with different times, only one period each PASS 9500,93600 AddPeriod1 Checks to see if the new period is added to an empty timetable PASS 9500,93600 PAS	Checks to see if it merges 2 timetables together mergeTimetable1 Checks the merging of and empty timetable and filled timetable PASS 1500,93500197 mergeTimetable2 Checks the merging of 2 tables with contradicting times PASS 3500,93500 1900,93500197 mergeTimetable3 Checks the merging of 2 tables with contradicting times PASS 9000,93500197 293500,93500197 mergeTimetable4 Checks the merging of 2 tables with outradicting times PASS 9000,93500197 393500,93500197 mergeTimetable5 Checks the merging of 2 tables with different times PASS 9000,93500197 mergeTimetable6 Checks the merging of 2 tables with different times PASS 9000,93500197 mergeTimetable6 Checks the merging of 2 tables with different times, only one period each PASS 9000,93500197 mergeTimetable6 Checks the merging of 2 tables with different times, only one period each PASS 9000,935000 21600,43200 mergeTimetable6 Checks the merging of 2 tables with different times, only one period each PASS 9000,935000 21600,43200 mergeTimetable6 Checks to see if the new period is added to an empty timetable PASS 9000,935000 21600,43200 Add period2 Checks to see if the new period is added to an empty timetable PASS 9000,935000 21600,4350	Checks to see if it merges 2 timetables together merge Timetable1 Checks the merging of 2 ables with contradicting times PASS 9350,97300 9000,9360019720,1006001216 merge Timetable2 Checks the merging of 2 tables with contradicting times PASS 9350,97300 9000,9360019720,1006001216 merge Timetable3 Checks the merging of 2 tables with contradicting times PASS 9300,935001972 93500,97300121600,43200 merge Timetable4 Checks the merging of 2 tables with different times, only one period each PASS 9000,936001972 93500,97300121600,43200 merge Timetable5 Checks the merging of 2 tables with different times, only one period each PASS 9000,93600 21600,43200 merge Timetable6 Checks the merging of 2 tables with different times, only one period each PASS 9000,93600 21600,43200 merge Timetable6 Checks to a timetable Period 1 Period 2 9000,93600 21600,43200 merge Timetable6 Checks to a timetable PASS 9000,93600 21600,43200 merge Timetable6 Checks to a timetable PASS 9000,93600 21600,43200 merge Timetable6 Checks to a timetable PASS 9000,93600 21600,43200 merge Timetable6 Checks to a timetable PASS 9000,93600 21600,43200 merge Timetable6 Checks to a timetable PASS 9000,93600 21600,43200 merge Timetable6 Checks to a timetable PASS 9000,93600 21600,43200 merge Timetable6 Checks to a timetable PASS 9000,93600 21600,43200 merge Timetable6 Checks to a timetable PASS 9000,93600 21600,43200 merge Timetable6 Checks to a timetable PASS 9000,93600 21600,43200 merge Timetable6 Checks to a timetable PASS 9000,93600 21600,43200 merge Timetable6 PASS 9000,93600 21600,43200 merge Timetable6 Checks to a timetable PASS 9000,93600 21600,43200 merge Timetable7 PASS 9000,93600 21600,43200 merge Timetable8 PASS 9000,93600 21600,43200 merge Tim	Checks to see if it merges 2 timetables together PASS Firmy 90000,33600 7200,100600 21600,43200 PEmpty 90000,33600 7200,100600 PEMPTY 90000,33600 7200 PEMPTY 900000,33600 7200 PEMPTY 90000,33600 7200 PEMPTY 90000,33600 7200 PEMP	Checks to see if it merges 2 timetables together PASS Timetable 1 Timetable 2 PASS PEmpty 9000,93600 97200,10600(21600,43200 PASS P