Test Class	Function Name	Test ID	Description	Status	Test Data 1	Test Data 2	Test Data 3	Test Data 4	Test Data 5	Expected output
Controller	**login()**		**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
		customerlogin04	The username is right and password is right, should return the user	PASS	admin	password				USER
Controller	**login()**		**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
		ownerlogin04	The username is right and password is right, should return the user	PASS	Ownertest	1234				OWNER
Controller	**register()**		**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
		customerregister03	All details are valid but the name, returns null	PASS	admin	password	Russell1234	12 Melbourne	03987656789	NULL
		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
Controller	**addEmployee()**		**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
SQLiteConnection	**getDBConnection()**	**testConnection**	**Checks if it can connect to the database**	**PASS**						
SQLiteConnection	**createCustomer()**	**testCreateCustomer**	**Checks if it will create a customer in the database**	**PASS**						
SQLiteConnection	**createEmployee()**	**testEmployeeFunction**	**creates some employees and check if they are created in the database**	**PASS**						
Peroid	**convertSecondsToDay()**		**Checks the conversion of seconds to days**		**Seconds**					
		testConvertSecondsToDay1	Checks if it converts to tuesday	PASS	86401					TRUE
		testConvertSecondsToDay2	Checks if it converts to monday	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 Monday	PASS	1					TRUE
Peroid	**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
Peroid	**checkIsValidWeekday()**									
		checkValidDay1	Checks if "Monday" is a valid date string	PASS						TRUE
		checkValidDay2	Checks if "Sunday" is a valid date string	PASS						TRUE

Test Class	Function Name	Test ID	Description	Status	Test Data 1	Test Data 2	Test Data 3	Test Data 4	Test Data 5	Expected output
		checkValidDay3	Checks if "WEDNESDAY" is a valid date string	PASS						TRUE
		checkValidDay4	Checks if "friday" is a valid date string	PASS						TRUE
		checkValidDay5	Checks if "MON" is a valid date string	PASS						FALSE
		checkValidDay6	Checks if "Saturnday" is a valid date string	PASS						FALSE
Peroid	get24HrTimeFromWeekTime()									
		testConvertTimeTo24hrTime1	Check if 60 seconds is equal to the 24 hour time string "0:01"	PASS						TRUE
		testConvertTimeTo24hrTime2	Check if 3600 seconds is equal to the 24 hour time string "1:00"	PASS						TRUE
		testConvertTimeTo24hrTime3	Check if 86399 seconds is equal to the 24 hour time string "23:59"	PASS						TRUE
		testConvertTimeTo24hrTime4	Check if 86340 seconds is equal to the 24 hour time string "23:59"	PASS						TRUE
Peroid	combineWith()				Period 1 times	Period 2 times	Period 3 times			
		testCombineWith1	Check if period 1 combined with period 2 = period 3	PASS	5am-6am	6am-7am	5am-7am			TRUE
		testCombineWith2	Check if period 1 combined with period 2 = period 3	PASS	3.45am-6am	5am-11am	3.45am-11am			TRUE
Timetable	mergeTimetable()		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|97200,100600|21600,43200				TRUE
		mergeTimetable2	Checks the merging of 2 tables with contradicting times	PASS	93500,97300	90000,93600|97200,100600|21600,43200				TRUE
		mergeTimetable3	Checks the merging of 2 tables with contradicting times	PASS	90000,93600|972	93500,97300|21600,43200				TRUE
		mergeTimetable4	Checks the merging of 2 tables with different times	PASS	21600,43200	90000,93600|97200,100600				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
Timetable	addPeriod()		Adds periods to a timetable		Period 1	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
Timetable	addPeriod()		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|40000,43600			TRUE
		removePeriod2	Checks to see if the period is romoved from the timetable	PASS	40000,43600		90000,93600|40000,43600			TRUE
		removePeriod3	Checks to see if the all periods are romoved from the timetable	PASS	90000,93600	40000,43600	90000,93600|40000,43600			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
Timetable	toStringArray()		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|97200,100600|21600,43200					TRUE
Customer	getPermissions()		Checks the retreval of the boolean permissions array		Customer	Permissions				
		testGetPermissions1	Check the permissions of a customer against a false set of permissions	PASS	Test Customer	False Perms				FALSE
		testGetPermissions2	Check the permissions of a customer against a correct set of permissions	PASS	Test Customer	Correct Perms				TRUE
Uitlity	searchUser()				Username					
		testSearchUser1	Check if the user does not exist in the database will return null	PASS	Ownertest123					null
		testSearchUser2	Check if Owner exists in the database and login	PASS	Ownertest					OWNER

Test Class	Function Name	Test ID	Description	Status	Test Data 1	Test Data 2	Test Data 3	Test Data 4	Test Data 5	Expected output