Petroleum Company System Project Description:

Aim: To increase the efficiency of operation in the company

This system will be used for time scheduling and task distribution and operation management. This system will not be fully involved in the production process.

List of team members:

Project Manager: Ali Ehab Ali Hassan   21100800

Team Members:

Mohamed Ashraf Rabie Mohamed   21100854

Ramez Mohamed Fathi Ahmed   21100793

Youssef Khaled Abdelsalam Mohamed   21100865

Mohamed Ashraf Mahmoud Mohamed   21100873

Managing Tool Requests by Engineer

Test requirements:

1.1-ensure that Engineers have the send tool request option.

1.2-ensure that there are two options to send a request.

1.3-ensure that if the engineer clicks upload document he is asked to drag and drop a document or select it from his drive.

1.4-ensure that if the engineer clicks the type of document, he will be prompted to enter his request via keyboard

1.5-ensure that the engineers' uploaded document is sent correctly to the Industry Development Department

1.6-ensure that the typed request is sent correctly to Industry Development Department

1.7-ensure that if the engineers request is approved by the Industry Development Department it is sent to the Financial Department

1.8-ensure that if the engineers request is disapproved by the Industry Development Department it is not sent to the Financial Department

1.9-ensure that if the Industry Development Department disapproves the Engineers request the Engineer receives a response that the request has been declined by the Industry Development Department

1.10-ensure that if the Financial Department disapproves the Engineers request the Engineer receives a response that the request has been declined by the financial department

1.11-ensure that if the financial department approves the Engineers' request that he is sent a response informing him of the approval of both departments.

1.12-Ensure that the Engineer can see the approval response.

1.13-Ensure that the Engineer can see the disapproval response by the Industry Development Department

1.14-Ensure that the Engineer can see the disapproval response by the Financial Development Department

Managing Employee Requests

Test requirements:

2.1-ensure that Engineers have the send tool request option.

2.2-ensure that there are two options to send a request.

2.3-ensure that if the engineer clicks upload document he is asked to drag and drop a document or select it from his drive.

2.4-ensure that if the engineer clicks the type of document, he will be prompted to enter his request via keyboard

2.5-ensure that the engineers' uploaded document is sent correctly to the Industry Development Department

2.6-ensure that the typed request is sent correctly to Industry Development Department

2.7-ensure that if the engineers request is approved by the Industry Development Department it is sent to the Financial Department

2.8-ensure that if the engineers request is disapproved by the Industry Development Department it is not sent to the Financial Department

2.9-ensure that if the Industry Development Department disapproves the Engineers request the Engineer receives a response that the request has been declined by the Industry Development Department

2.10-ensure that if the Financial Department disapproves the Engineers request the Engineer receives a response that the request has been declined by the financial department

2.11-ensure that if the financial department approves the Engineers' request that he is sent a response informing him of the approval of both departments.

2.12-Ensure that the Engineer can see the approval response.

2.13-Ensure that the Engineer can see the disapproval response by the Industry Development Department

2.14-Ensure that the Engineer can see the disapproval response by the Financial Development Department

Display

Test requirements:

3.1-check if the display button is available.

3.2-check if the display shift time button is available.

3.3-check if the display transportation time button is available.

3.4-check if the display salary button is available.

3.5-check if the display required tasks button is available.

3.6-check that the display shift time displayed successfully.

3.7-check if the shift time is edited the display shift time is displayed successfully.

3.8-check that the display transportation displayed successfully.

3.9-check if the transportation time is edited and the display shift time is displayed successfully.

3.10-check that the display salary displayed successfully.

3.11-check if the salary is edited (increased due to promotion or decreased due to a fine)

3.12-check that the display task option is displayed correctly.

3.13-check that at editing the tasks for the employee it is updated correctly.

Managing Insurance

Test requirements:

4.1-Every employee does not exceed the limited number of dependents-Ensuring that the employee is still working for the company.

4.2-Ensuring that the health insurance fees are taken off from the employees' salary

4.3-Ensuring that if the number of dependents increased the health insurance fees have been increased

4.4-Ensuring that if an employee uploaded an insurance request that it will be added to report list that will be reviewed

4.5-Ensuring that if the request has been approved, the approvement is sent to the employee.

Manage Maintenance Request

Test requirements:

5.1-Check that send employee request option is available

5.2-Check that employee can upload a document

5.3-Check that the employee has the uploaded document sent successfully

5.4-Check that the employee has the option to type a document (the malfunctioned machine)

5.5-Check that the problem request was sent correctly

5.6-Check that the malfunctioned machine is edited in the database as out of service

5.7-Check that all related employees see the machine as malfunctioned

5.8-Check that the maintenance department receives the workers confirmation correctly

5.9-Check that the notification that the maintenance is underway is sent to the requester correctly

5.10-Check that the maintenance department receives the workers repair confirmation correctly

5.11-Check that after repair the machine is edited in the database as back in operation-Check that the requestor and all related employees receive confirmation of the repair

	M1	M2	M3	M4
Employee	1	1	1	1
Department	1	1	1	1
Request		1	1	
Schedual			1	
Report				1
Company Manager		1		

M1: Login

M2: Day Off request

M3: Maintenance Request

M4: Upload Report

Integration testing will be done with the top bottom approach.

UserName   Enter UserName

Password   Enter Password

Position   ▽

Login

M1	lOGIN
1	LOGIN
1.1	UseName --> Text Field
1.1.1	must contain an @ symbol
1.1.2	must ed with .com or .net or any valid top-level domain
1.2	Password --> Text Field
1.2.1	password must be min 8 characters long
1.2.2	password must con 1 uppercase letter
1.2.3	password must contain at least 1 number
1.2.4	password must contain at least 1 Letter
1.2.5	password must contain at least 1 special character
1.3	Position --> drop down Box
1.4	Login --> button

Test Requirement		Test Case Number		Component		Input Values			Description		
With Vaild Input		1				Mohamed@gmail.com			Accepted		
With Invalid input		2				mohamed.com			Error Message --> it must contain an@symbol		
		3		UserName		mohamed@			Error Message --> it must contain a valid top level domain		
		4				mohamed			invalid as it must conatin an @ symbol or a valid to level domain		
With Valid Input		5				123456@Ab			Accepted		
With Invalid Input		6				1@aA			Error Message --> password is invalid		
		7				1@abcdefghij			Error Message --> password is invalid		
		8		Password		Abcdefghij@			Error Message --> password is invalid		
		9				1234567@23			Error Message --> password is invalid		
		10				1234Abcdef			Error Message --> password is invalid		

Test Requirement		Test Case Number		Component		Action			Description		
With Valid Input		11				on click			A list of all the Positions apperar(Engineer, Worker ,Lab Technician, Security ,etc)		
		12		Position		on click			no positions appear		

With Valid Input	13		username --> Mohamed@gmail.com password --> 123456@Ab then click the button	Accepted
	14		username --> Mohamed@gmail.com password --> 123456@Ab then click the button	Error Message --> Invalid Credentials
With Invalid input	15		username --> mohamed.com password --> 123456@Ab then click the button	Error Message --> Invalid UseName
	16		username --> mohamed@ password --> 123456@Ab then click the button	Error Message --> Invalid UseName
	17		username --> mohamed password --> 123456@Ab then click the button	Error Message --> Invalid UseName
	18	Login	username --> Mohamed@gmail.com password --> 123456@Ab then click the button	Error Message --> Invalid Password
	19		username --> Mohamed@gmail.com password --> 1@aA then click the button	Error Message --> Invalid Password
	20		username --> Mohamed@gmail.com	Error Message --> Invalid Password

		password --> 1@abcdefghij then click the button	
21		username --> Mohamed@gmail.com password --> Abcdefghij@ then click the button	Error Message --> Invalid Password
22		username --> Mohamed@gmail.com password --> 1234567@23 then click the button	Error Message --> Invalid Password
23		username --> mohamed.com password --> 1234Abcdef then click the button	Error Message --> Invalid Password and username

Request History

Day off Request   **Maintenance Request**

Requested date   /   /

Write reason

Submit

M2	Request Day Off
2	DayOff Request
2.1	Write Reqason --> Text Field
2.1.1	Must have min of 20 words
2.2	Request Date --> Text Field
2.2.1	Must only caontain numbers and /
2.2.2	must be writeen in a xx/xx/xxxx format(day/month/year)
2.3	Request History --> Text Box
2.4	Submit --> Button
2.4.1	both the request data and the reason text fields must be filled

Test Requirement	Test Case Number		Component		Input Values		Description
With Valid input	1				25/5/2023		Accepted
	2				20/5/2020		Error Message ---> the entered date has already Passed
With Invalid input	3		Request Date		t/5/2023		Error Message ---> the entered date must be written in intgers only
	4				5/25/2023		Error Message ---> the entered date must be written in DD/MM/YYYY
	5				Tuesday/june/2023		Error Message ---> the entered date must be written in intgers only

With Valid input	6				I cannot come because of having my son graduation at school and it is at the same time of work		Accepted
With Invalid input	7		Write Reason		I cannot come		Error Message ---> the entered reason must be more than 20 words

Test Requirement	Test Case Number		Component		Action		Description
With Valid input	8				Onclick && Requested Date ---> 25/5/2023 Write Reason--> I cannot come because of having my son graduation at school and it is at the same time of work		Accepted
With Invalid input	9				Onclick && Requested Date ---> 20/5/2023		Error Message ---> the entered date has already Passed

		Write Reason--> I cannot come because of having my son graduation at school and it is at the same time of work	
10		Onclick && Requested Date ---> t/5/2023 Write Reason--> I cannot come because of having my son graduation at school and it is at the same time of work	Error Message ---> the entered date must be written in intgers only
11	Submit	Onclick && Requested Date ---> 5/25/2023 Write Reason--> I cannot come because of having my son graduation at school and it is at the same time of work	Error Message ---> the entered date must be written in DD/MM/YYYY
12		Onclick && Requested Date ---> Tuesday/june/2023 Write Reason--> I cannot come because of having my son graduation at school and it is at the same time of work	Error Message ---> the entered date must be written in intgers only
13		Onclick && Requested Date ---> 25/5/2023 Write Reason--> I cannot come	Error Message ---> the entered reason must be more than 20 words
14		Onclick &&	Error Message -->

		Requested Date ---> Blank Write Reason--> I cannot come because of having my son graduation at school and it is at the same time of work	Request Date Cannot be left Empty
15		Onclick && Requested Date ---> 25/5/2023 Write Reason--> Blank	Error Message --> Write Reason Cannot be left Empty
16		Onclick && Requested Date ---> Blank Write Reason--> Blank	Error Message --> Request Date Cannot be left Empty, Write Reason Cannot be left Empty

Request History

Day off Request

Maintenance Request

Location

Write reason

Submit

3	Request Maintenance
3.1	Request History --> Text Box
3.2	Location --> Text Field
3.2.1	must not have punctuation marks
3.2.2	Location must be inside the company
3.3	Write reason --> Text Field
3.3.1	can't have less than 20 words
3.4	submit --> button
3.4.1	Both the loaction and reason must be filled

Test Requirement	Test Case Number	Component	Input Values	Description
With Valid Input	1		Department 101	Accepted
	2	Location	10th street in Alexandria	Accepted
With Invalid Input	3		Department @ 101	Error Message --> l mustn't contain pu

Test Requirement	Test Case Number	Component	Input Values	Description
With Valid Input	4		User inputs 20 or more words	Accepted
With Invalid Input	5	Write reason	I request a maintainance request on department 101	Error Message --> V reason must conta 20 words

Test Requirement	Test Case Number	Component	Action	Description
With Valid input	6		Location --> Department 101 , Write reason --> User inputs 20 or more words	Accepted
With Invalid Input	7		Location --> 10th street in Alexandria , Write reason --> User inputs 20 or more words	Error Message --> l must be inside the
	8	Submit	Location --> Department @ 101 , Write reason --> User inputs 20 or more words	Error Message --> l mustn't contain pu
	9		Location --> Department 101 , Write reason --> I request a maintainance	Error Message --> V reason must conta 20 words

		request on department 101	
	10	Location --> Department @ 101 , Write reason --> I request a maintainance request on department 101 request on department 101	Error Message --> V reason must conta 20 words, Location contain punctuatio
		Location --> Blank, Write reason --> User inputs 20 or more words	Error Message --> Location Cannot Be
		Location --> Department 101 , Write reason --> Blank	Erorr Message --> reason cannot be E
		Location --> Blank Write reason --> Blank	Error Message --> Location Cannot Be ,reason cannot be

3		Request Maintenance						
3.1		Request History --> Text Box						
3.2		Location --> Text Field						
3.2.1		must not have punctuation marks						
3.2.2		Location must be inside the company						
3.3		Write reason --> Text Field						
3.3.1		can't have less than 20 words						
3.4		submit --> button						
3.4.1		Both the loaction and reason must be filled						

Test Requirement		Test Case Number		Component		Input Values		Description	
with valid inputs		1				report.pdf		Accepted	
with invalid inputs		2		Drag a report		report.xlx		Error message--> incorrect format must be pdf	
		3				report.png		Error message--> incorrect format must be pdf	

Test Requirement		Test Case Number		Component		Action		Description	
with valid inputs		4		Upload		uplaod report.pdf then click button		Accepted	
with invalid inputs		5				upload nothing click button		Error message--> empty reports box	

Unit Testing

U1

Employee		
Method	**Input**	**Expected Output**
Employee	id = 1, first_name = "John", last_name = "Doe", hireDate = "2022-01-01", salary = 50000, phoneNumber = {"555-1234"}, email = "johndoe@example.com", ssn = "123-45-6789", address = "123 Main St, Anytown, USA", birthDate = "1990-01-01", gender = "Male", Location location	Creates an Employee object with the specified values
SendEmployeeRequest	Request request object with requestID = 123, int departmentID = 1	Returns an int representing the request ID
SendMaintenanceRequest	Request request object with requestID = 456, int departmentID = 2	Returns an int representing the request ID
setID	int id = 2	Sets the employee's ID to 2
setFirstName	String firstName = ""Jane""	Sets the employee's first name to ""Jane""
setLastName	String lastName = ""Smith""	Sets the employee's last name to ""Smith""
setHireDate	String hireDate = ""2022-05-01""	Sets the employee's hire date to ""2022-05-01""
setSalary	float salary = 60000	Sets the employee's salary to 60000
addPhoneNumber	String phoneNumber = ""555-5678""	Adds a phone number to the employee's phone number array
removePhoneNumber	int index = 0	Removes the first phone number from the employee's phone number array
setEmail	String email = ""janesmith	Sets the employee's email address to ""janesmith@example.com""
setGender	String gender = ""Female""	Sets the employee's gender to ""Female""
setSSN	String ssn = ""987-65-4321""	Sets the employee's SSN to ""987-65-4321""
setAddress	String address = ""456 Elm St, Anytown, USA""	Sets the employee's address to ""456 Elm St, Anytown, USA""
setBirthDate	String birthDate = ""1995-05-01""	Sets the employee's birth date to ""1995-05-01""
setMainLocation	Location location	Sets the employee's main location
setSchedule	Schedule schedule	Sets the employee's schedule

U2

Request

Method	Input	Expected Output
Request	ID = 1, description = "Fix broken printer", employeeID = 2	Creates a Request object with the specified values
setID	int ID = 3	Sets the request ID to 3
setDescription	String description = ""Replace broken lightbulb""	Sets the request description to ""Replace broken lightbulb""
setEmployeeID	int employeeID = 4	Sets the employee ID of the request to 4
setDepartmentID	int departmentID = 5	Sets the department ID of the request to 5

U3

Company Manager

Method	Input	Expected Output
setFirstName	String firstName = "Jane"	Sets the employee's first name to "Jane"
setLastName	String lastName = "Doe"	Sets the employee's last name to "Doe"
setHireDate	String hireDate = "2022-02-01"	Sets the employee's hire date to "2022-02-01"
setSalary	float salary = 60000.0	Sets the employee's salary to 60000.0
addPhoneNumber	String phoneNumber = "555-555-5555"	Adds a phone number to the employee's contact information
removePhoneNumber	int index = 0	Removes the first phone number from the employee's contact information
setEmail	String email = "janedoe@example.com"	Sets the employee's email address to "janedoe@example.com"
setGender	String Gender = "Female"	Sets the employee's gender to "Female"
setSSN	String ssn = "987-65-4321"	Sets the employee's Social Security Number to "987-65-4321"
setAddress	String address = "456 Main St"	Sets the employee's address to "456 Main St"
setBirthDate	String birthDate = "1995-08-15"	Sets the employee's birth date to "1995-08-15"
ValidateHiringRequest	int requestID = 1	The status of the specified hiring request is set to "Approved"
ValidatePurchaseRequest	int requestID = 2	The status of the specified purchase request is set to "Approved"
ValidateToolPurchaseRequest	int requestID = 3	The status of the specified tool purchase request is set to "Approved"

U4

Report		
Method	**Input**	**Expected Output**
Report	ID = 1 , description = "Monthly Sales Report", employeeID = 2, departmentID = 3 , URL = "http://example.com/report.pdf" , PDF[] pdf	Creates a Report object with the specified values
setID	int ID = 4	Sets the report ID to 4
setDescription	String description = ""Quarterly Sales Report""	Sets the report description to ""Quarterly Sales Report""
addPDF	PDF data object	Adds a PDF object to the report
removePDF	int PDF_Index = 0	Removes the first PDF object from the report
addURL	String url = ""http://example.com/report2.pdf""	Adds a URL to the report
removeURL	int URL_Index = 1	Removes the second URL from the report
setEmployeeID	int employeeID = 5	Sets the employee ID associated with the report to 5 with the report to 5
setDepartmentID	int departmentID = 6	Sets the department ID associated with the report to 6

U5

Schedule		
Method	**Input**	**Expected Output**
addTask	Task task object with taskID = 1, description = "Prepare monthly report" , employeeID = 2, departmentID = 3, priority = 2, status = "Inprogress", date = "2022-05-23", and String date = "2022-05-23"	The Task object is added to the schedule for the specified date
setShiftStartTime	String time = ""08:30""	Sets the start time of the shift to ""08:30""
setShiftEndTime	String time = ""17:30""	Sets the end time of the shift to ""17:30""

U6

Department		
Method	**Input**	**Expected Output**
sendHiringRequestToHR(Request request , int departmentID)	`Request request` object with `requestID = 123`, `int departmentID` = 1	HR department receives the hiring request with `requestID = 123` from department with `departmentID = 1`
addEmployee(int employeeID)	`int employeeID` = 1	Employee with `employeeID = 1` is added to the department
removeEmployee(int employeeID)	`int employeeID` = 1	Employee with `employeeID = 1` is removed from the department
promoteEmployee(int employeeID)	`int employeeID` = 1	Employee with `employeeID = 1` is promoted
sendEmployeeRequestValidation (int employeeID, int requestID)	`int employeeID` = 1, `int requestID` = 456	Employee with `employeeID = 1` sends request with `requestID = 456` for validation
sendPurchaseRequestToFinancialDepartment (int employeeID, int requestID)	`int employeeID` = 1, `int requestID` = 789	Financial department receives purchase request with `requestID = 789` from employee with `employeeID = 1`
displayHiringRequestValidation(int requestID)	`int requestID` = 123	Hiring request validation with `requestID = 123` is displayed
setDepartmentID(int departmentID)	`int departmentID` = 2	Sets the department ID to `2`
addLocation(int locationID)	`int locationID` = 3	Adds a location with `locationID = 3` to the department
removeLocation(int locationID)	`int locationID` = 3	Removes the location with `locationID = 3` from the department
addTaskToEmployee(Task task, int employeeID)	`Task task` object with `taskID = 1` and `description = "Complete project report"`, `int employeeID` = 1	Employee with `employeeID = 1` is assigned the task with `taskID = 1` and `description = "Complete project report"`

Test Plan

Unit Test U1

Unit test U2

Unit Test U3

Unit test U4

Unit Test U5

Unit test U6

Unit test M1

Unit test M2

Unit test M3

Unit test M4

Integration Testing for M1, U6, U1

Regression testing for U6

Regression testing for U1

Regression Testing for M1, U1, U6

Integration Testing for M2, U6, U3, U2, U1

Regression testing for U6

Regression testing for U3

Regression testing for U2

Regression testing for U1

Regression Testing for M2, U6, U2, U3, U1

Regression Testing for M1, U1, U6

Integration Testing for M3, U6, U5, U2, U1

Regression testing for U6

Regression testing for U5

Regression testing for U2

Regression testing for U1

Regression Testing for M3, U6, U5, U2, U1

Regression Testing for M1, U1, U6

Regression Testing for M2, U6, U2, U3, U1

Integration Testing for M4, U6, U4, U1

Regression testing for U6

Regression testing for U4

Regression testing for U1

Regression Testing for M4, U6, U4, U1

Regression Testing for M3, U6, U5, U2, U1

Regression Testing for M1, U1, U6

Regression Testing for M2, U6, U2, U3, U1

M1 is dependent on U1, U6

M2 is dependent on U6, U3, U2, U1

M3 is dependent on U6, U5, U2, U1

M4 is dependent on U6, U4, U1

Chart Title

Task	Timeline
Unit test U4	
Unit test U6	
Unit test M2	
Unit test M4	
Regression testing for U6	
Regression Testing for M1 U1 U6	
Regression testing for U6	
Regression testing for U2	
Regression Testing for M2 U6 U2 U3 U1	
Integration Testing for M3 U6 U5 U2 U1	
Regression testing for U5	
Regression testing for U1	
Regression Testing for M1 U1 U6	
Integration Testing for M4 U6 U4 U1	
Regression testing for U4	
Regression Testing for M4 U6 U4 U1	
Regression Testing for M1 U1 U6	

Timeline axis: 5/25/2023, 6/4/2023, 6/14/2023, 6/24/2023, 7/4/2023, 7/14/2023, 7/24/2023, 8/3/2023, 8/13/2023, 8/23/2023