Vulture Services Class Diagram

JobController
+jobField: String
+jobNumber: Int
+clientID: String
+arrivalDate: Date
+returnDate: Date
+quotedParts: String
+motorID: String
+manufacturer: String
+manYear: String
+labourTimeValue: String
+labourTime: int
+checkedBy: String
+checkedDate: Date
+jobStatus: String
+InspectedBy: String
+inspectedDate: Date
+approved: Boolean
+connection: connection
+ps: PreparedStatement

+loadJob(jobNumber: int)
+newJob()
+save(ae: ActionEvent)
+saveTask()
+createTask(taskName: String, taskNotes: String, taskAssigned: String, taskComplete: Boolean)
+formatTextField()
+setInspectedBy()
+setCheckedBy()
+sqlConfirmation()
+loadTask(jobNumber: int)
+displayTasks()
+suspendTask(ae: ActionEvent)
+createTask()
+exportJob(ae: ActionEvent)
+checkRow()
+completeTask(ae: ActionEvent)
+navigate(event: ActionEvent)
+checkJob(jobNumber: int)
+clearField(ae: ActionEvent)
+loadMenuScreen(ae: ActionEvent)
+calculateLabourTime(key: KeyEvent)
+checkInputErrors()
+initialiseArrays()
+confirmation()
+enterJobNumber(ae: ActionEvent)
+setEndDate(event: ActionEvent)
+clearFields()

Job
+client: String
+quotedParts: String
+jobNumber: int
+labourTime: int
+arrivalDate: Date
+returnDate: Date

+job(client: String, jobNumber: int, laboutTime: int, arrivalDate: Date, returnDate: Date)
+job(client: String, jobNumber: int, quotedParts: String, returnDate: Date)

NotificationsController
+con: Connection
+ps: PreparedStatement

+deleteNotification(alert: String)
+getNotifications()
+setEventHandlers()

UserModalController
+originalUsername: String
+userID: Integer
+con: Connection
+ps: preparedStatement

+getUserID()
+setUsername(username: String)
+getUserRoles()
+setUserRoles(type: String)
+editUser()
+deleteUser()
+setEditModal()
+setNewModal()
+close()
+saveUser()
+newUser()

Variables
+userID: int
+username: String

Part
+part: String
+available: String

+getPart()
+setPart(part: Sting)
+getavailable()
+setAvailable(availablle: String)

JobDelayController
+jobField1: String
+jobNumber: Int
+connection: Connection
+ps: preparedStatement

+loadJob()
+saveStatus()
+loadTaskStatus()
+saveTaskStatus()
+alertTechnician()
+alertManagment()
+alertCustomerServices()
+navigate(event: ActionEvent)

StatisticsController
+manufacturerList: ArrayList<String>
+durationChart: AreaChart<Number>
+jobDuration: Hashmap<String><Long>
+taskDuration: Hashmap<String><Long>
+estTaskDuration: Hashmap<String><Long>
+jobDelay: Hashmap<String><Integer>
+inspectionDelay: Hashmap<String><Integer>
+manufacturerInspectionList: ArrayList<String>
+motorList: ArrayList<String>
+manufacturer: String
+connection: Connection
+ps: preparedStatement

+getManufacturer()
+compareJobDuration()
+compareTaskDuration()
+compareEstimatedTaskDuration()
+unexpectedDelayFrequency()
+finalInspectionFailure()
+populateInspectionFailure()
+populateJobDuration()
+populateTaskDuration()
+populateEstTaskDuration()
+populateJobDelay()

LoginController
+username: String
+PasswordField: Password
+connection: Connection
+ps: prepareStatement

+login()

MenuController
+setJobScreen()
+setLoadJobScreen()
+setTaskScreen()
+setUserAccountScreen()
+setJobDelayScreen()

Main
+main(args[]: String)

Database
+driverName: String
+con: Connection

+getConnection()

UserAccountController
+con: Connection
+ps: preparedstatement

+navigate(event: ActionEvent)
+checkUserType()
+getSuspendedTask()
+getCurrentTask()
+getOverDueTasks()
+getActiveJobs()
+getOverdueJobs()
+getTotalCompletedJobs()
+getTotalActiveJobs()
+getTotalCompletedTaks()
+getTotalActiveTasks()
+getUsers()
+getJobsThatNeedParts()
+addPartsEventListeners()
+addUsersEventListeners()
+newUsers()
+orderMoreParts()
+orderNewParts()
+checkNotifications()
+showNotifications()

TaskController
+connection: Connection
+ps: PreparedStatement

+sortTaskName()
+sortCompleted()
+sortNotCompleted()
+sortUrgency()
+sortSuspended()
+sortDuration()
+sortAssigned()
+sortUnassigned()
+searchUsername()
+assignTask()
+fillTables()
+checkAssign()
+fillComboBoxes()
+setOverDue()
+fillTextFields()
+confirmation(message: String)
+navigate(ae: ActionEvent)

Keys
+databaseURL: String
+databaseUser: String
+databasePassword: String

SceneController
+screenMap: HashMap<String>
+mainScene: Scene

+start(primaryStage: Stage)
+stop()
+addScreen(name: String, pane: Pane)
+removeScreen(name: String)
+activate(name: String)
+registerScreens()
+SceneController(args[]: String)

Task
+taskName: String
+jobNumber: int
+description: String
+duration: int
+urgency: int
+Suspended: int
+Username: String
+complete: Boolean
+assignedTo: String
+taskNotes: String
+taskDuration: int
+taskComplete: Boolean
+taskUrgency: int
+taskSuspended: boolean
+dueDate: String
+assignedDate: String

+Task(taskName: String, jobNumber: int, taskNotes: String, taskDuration: int, taskComplete: boolean, taskUrgency: int, taskSuspended: boolean, assignedTo: String, dueDate: String, assignedDate: string)
+Task(taskName: String, jobNumber: int, taskNotes: String, taskDuration: int, taskComplete: boolean, taskSuspended: boolean, assignedTo: String)
+Task(taskName: String, jobNumber: int, taskNotes: String, taskDuration: int, taskComplete: boolean, taskUrgency: int, taskSuspended: boolean, assignedTo: String)
+Task(taskName: String, jobNumber: int, taskNotes: String, taskDuration: int, taskComplete: boolean, taskSuspended: boolean, assignedTo: String)
+Task(taskName: String, jobNumber: int, description: String, duration: int, urgency: int)
+Task(taskName: String, taskNotes: String, duration: int, assignedTo: String, complete: Boolean)
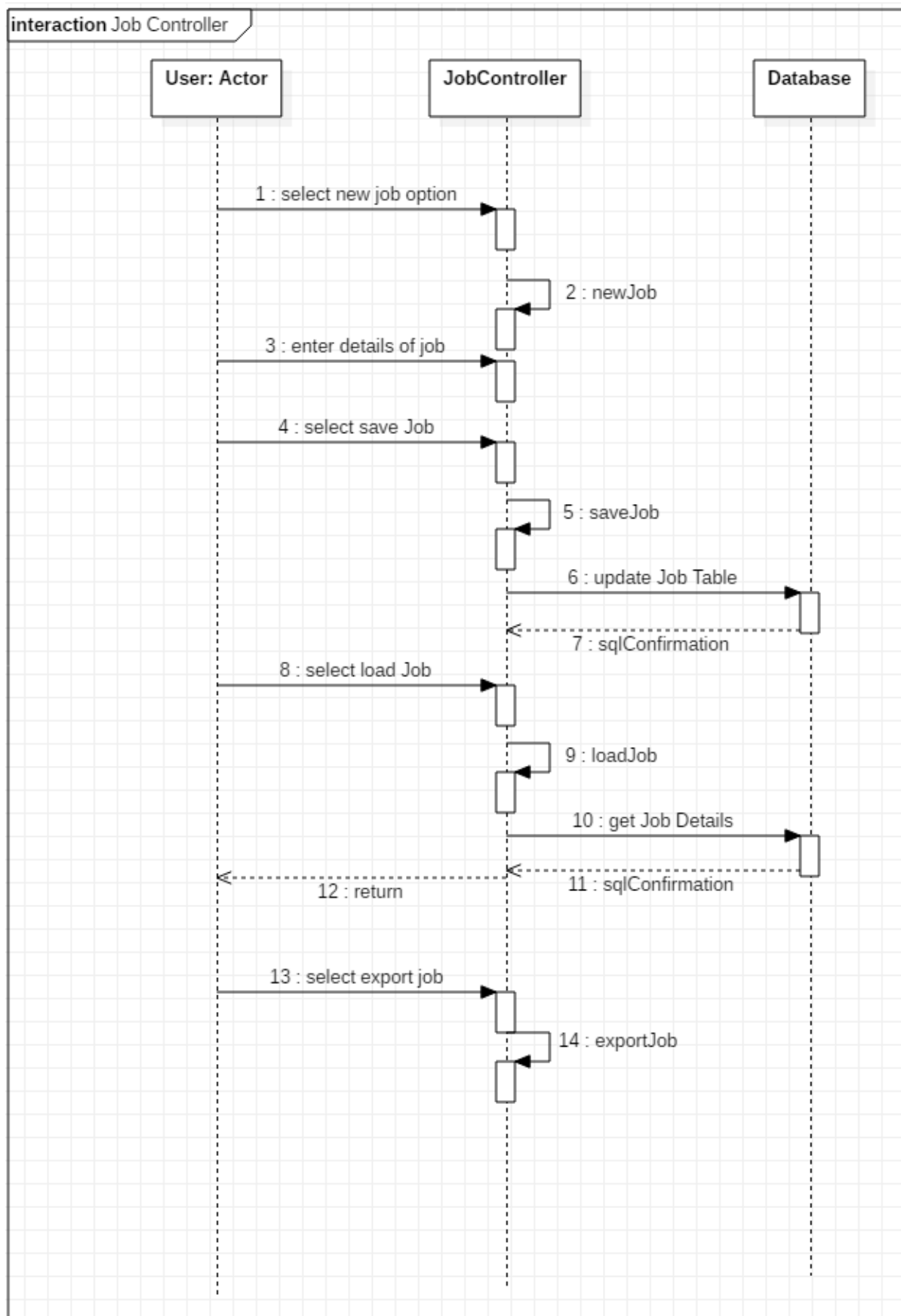
Vulture services Sequence Diagrams

Changing a job Status sequence diagram

interaction Change a Job or Task status

| User: Actor | JobDelayController | Database |

1 : enter in job number

2 : press load job button

3 : loadJob

4 : success

seq OPT

[if job status must be changed]

5 : change if active or suspended

6 : press save staus button

7 : save Status

8 : success

9 : confirmation message

10 : enter in a task ID

11 : press view task status button

12 : loadTaskStatus

13 : return

seq CombinedFragment1

[if task status needs changed]

14 : change if active or suspended

15 : press save staus button

16 : saveTaskStatus

17 : return

18 : confirmation message

Mission 3 – Matthew Pritchard – w16037991

Job controller sequence diagram

interaction Job Controller

User: Actor   JobController   Database

1 : select new job option

2 : newJob

3 : enter details of job

4 : select save Job

5 : saveJob

6 : update Job Table

7 : sqlConfirmation

8 : select load Job

9 : loadJob

10 : get Job Details

11 : sqlConfirmation

12 : return

13 : select export job

14 : exportJob

Assign a task sequence diagram
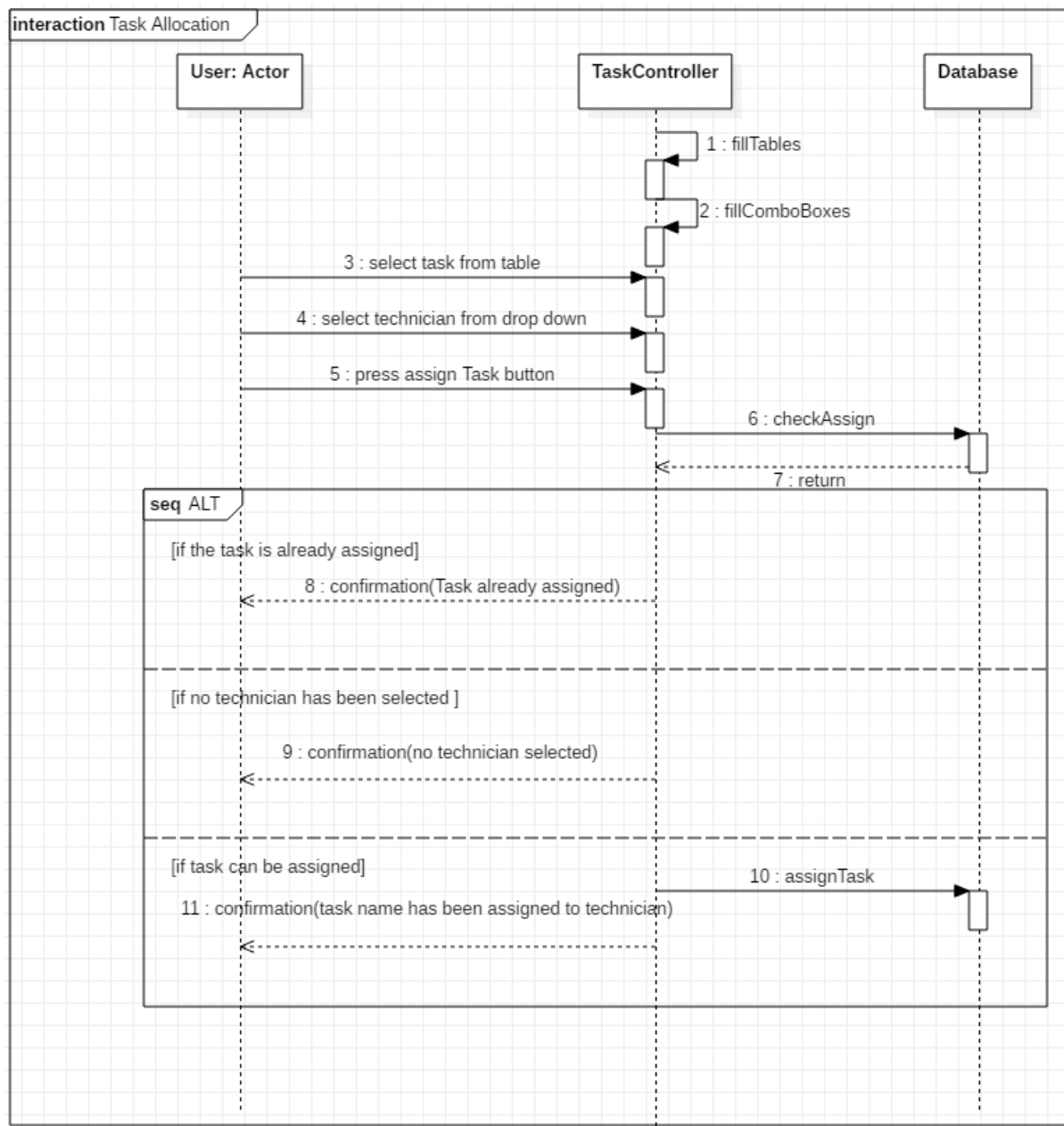
User account sequence diagram

interaction User Account Controller

| User: Actor | UserModalController | Database |

1 : enter user details and add
2 : newUser
3 : setUsername
4 : success
5 : setUserRoles
6 : success
7 : return
8 : select user to edit details
9 : editUser
10 : getUserID
11 : return
12 : setUsername
13 : success
14 : setUserRoles
15 : success
16 : success
17 : select user to delete
«destroy»
18 : deleteUser
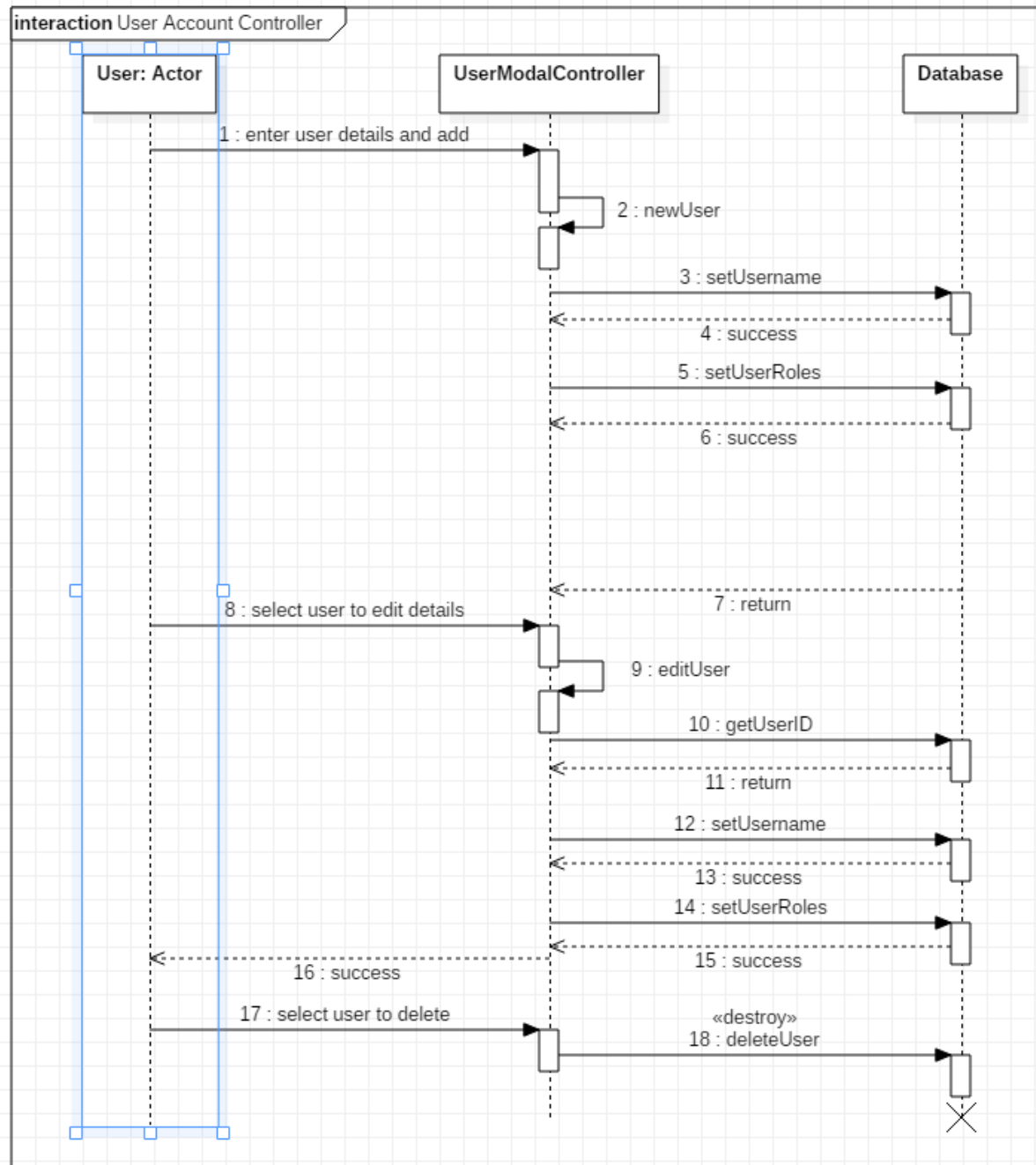
Sending alerts to users sequence diagram

Mission 3 – Matthew Pritchard – w16037991
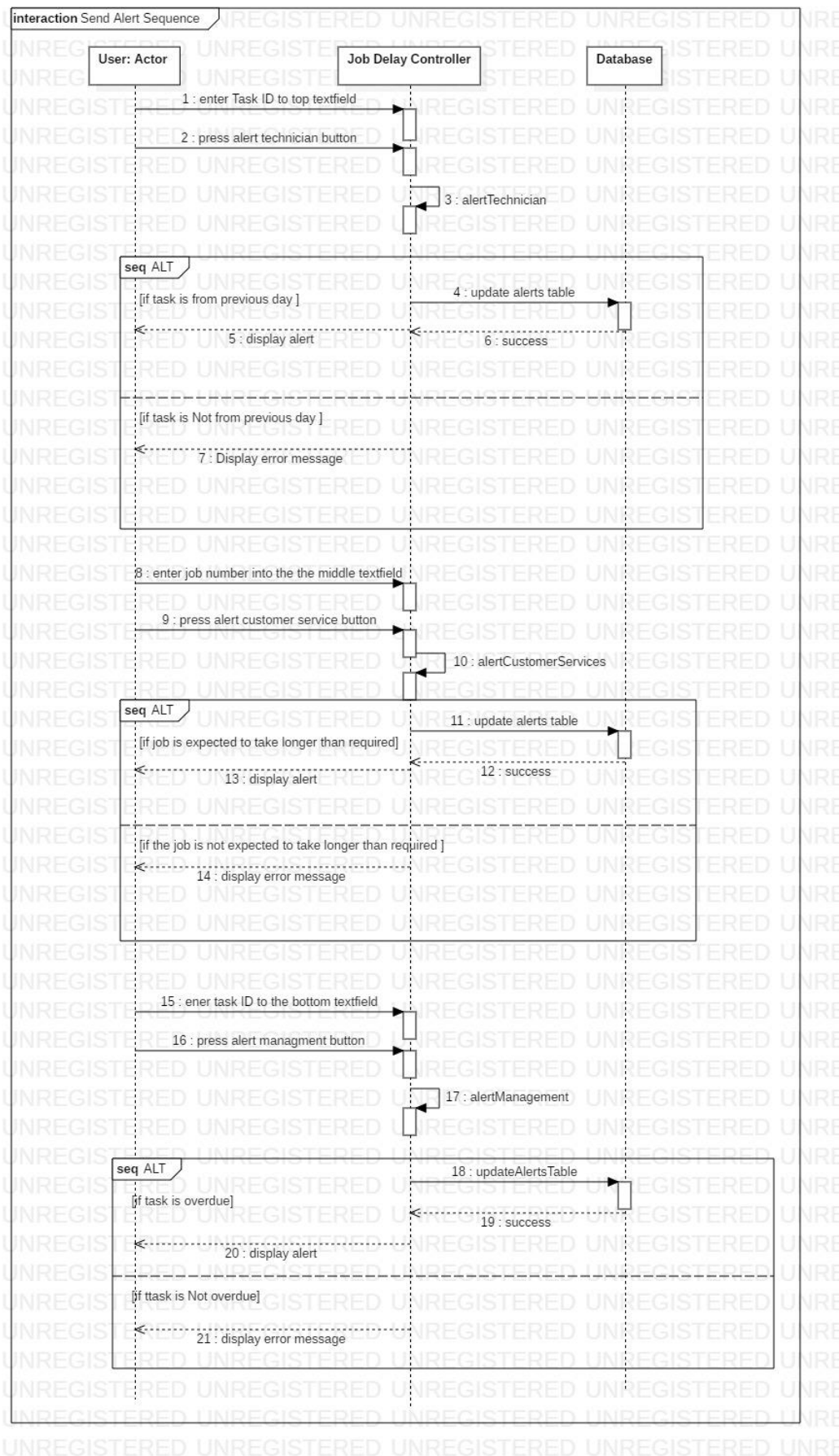
Logging on sequence diagram
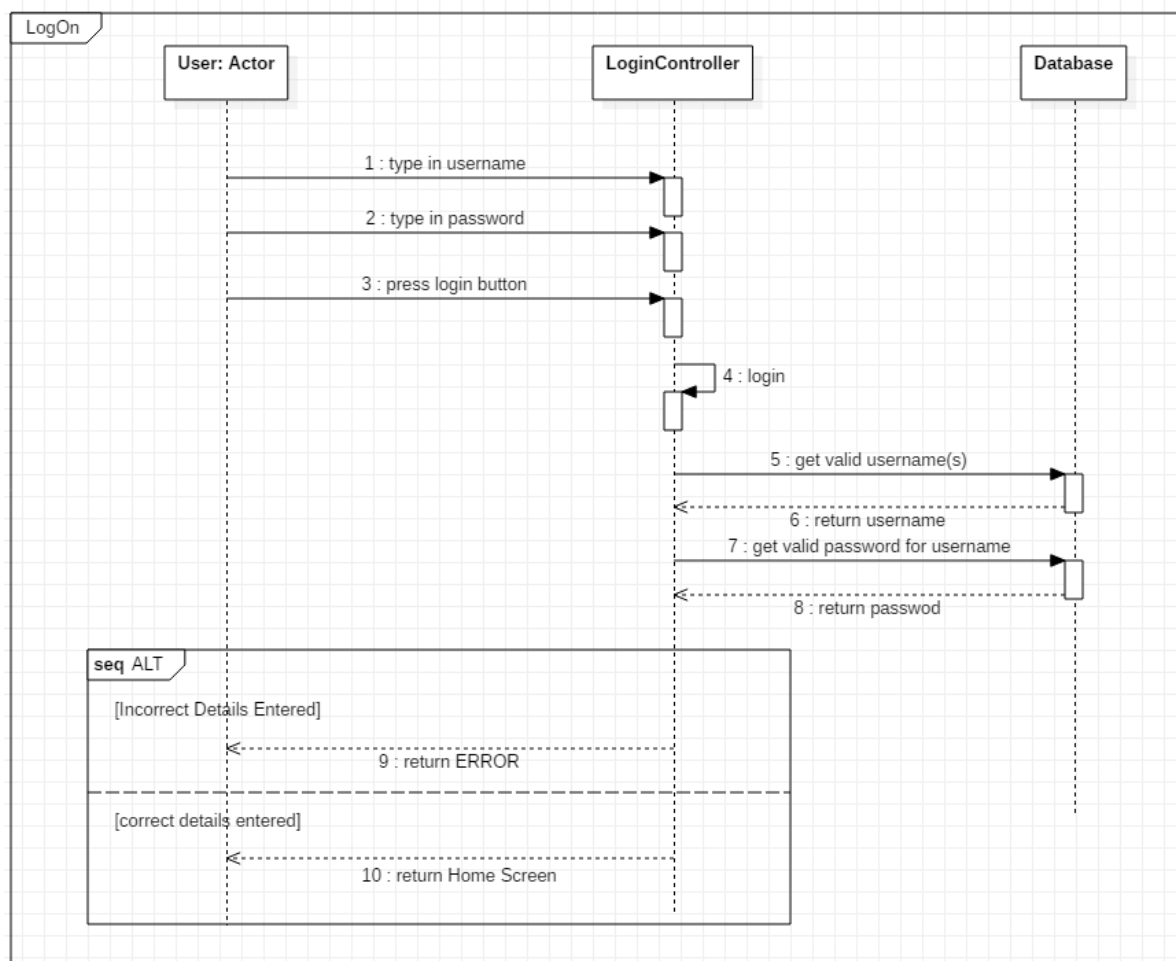
Filtering tasks on the task allocation page sequence diagram
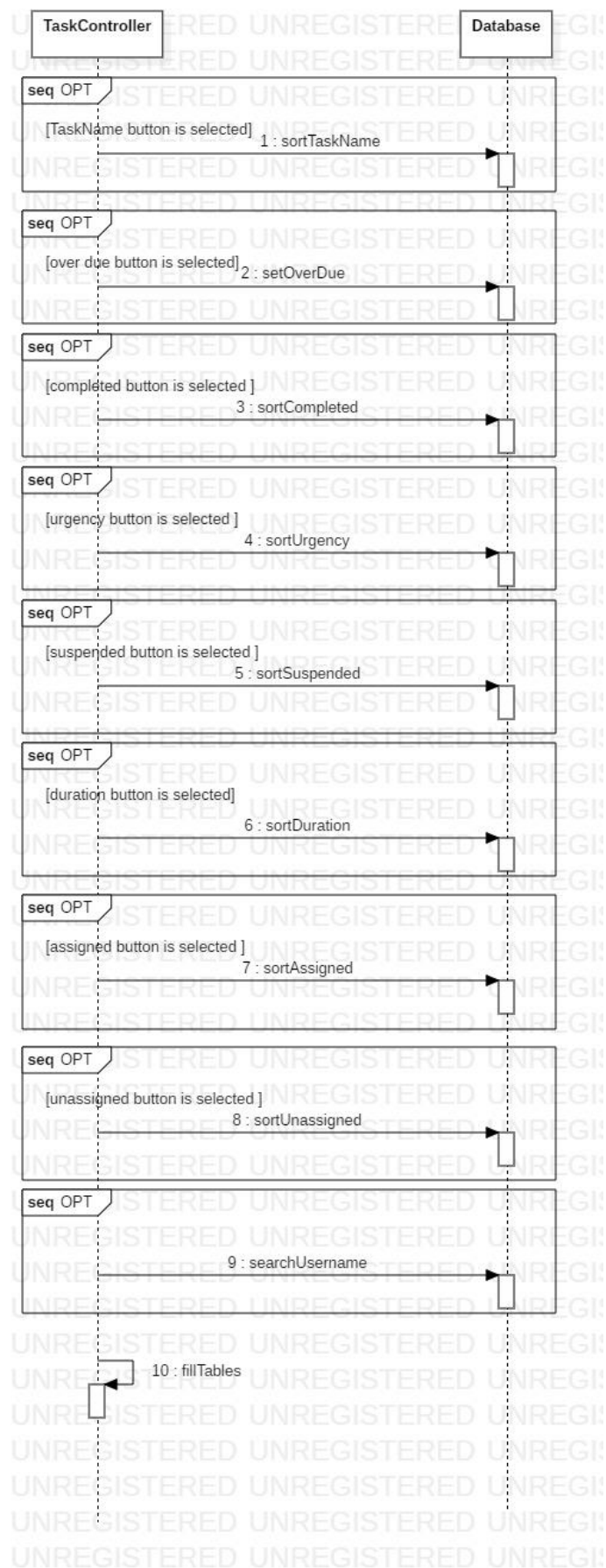
Vulture Services State Diagrams

Logging onto the system state diagram

display error message	Invalid Username or password

invalid

| Load System | Enter Username and Password | | valid | Valid Credentials | ● |
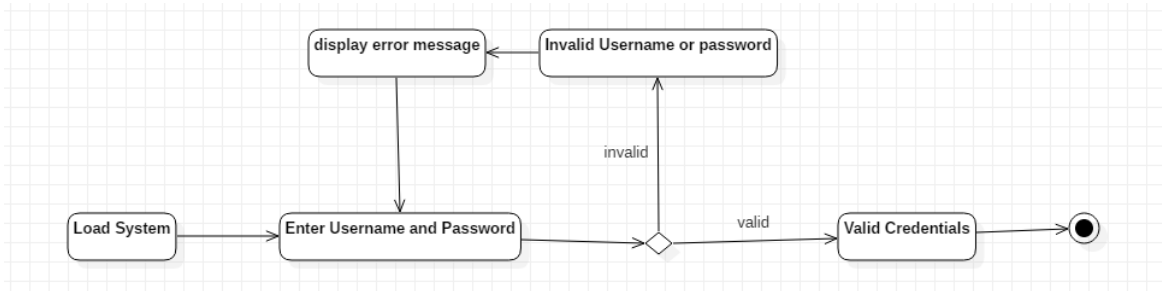
Task Allocation State diagram

Display error Message	task already assigned

invalid

| Select Task from List | Enter userame to assign task | Click assign task button | | valid | Task Assigned | ● |

invalid

| Display Error Message | invalid username entered |
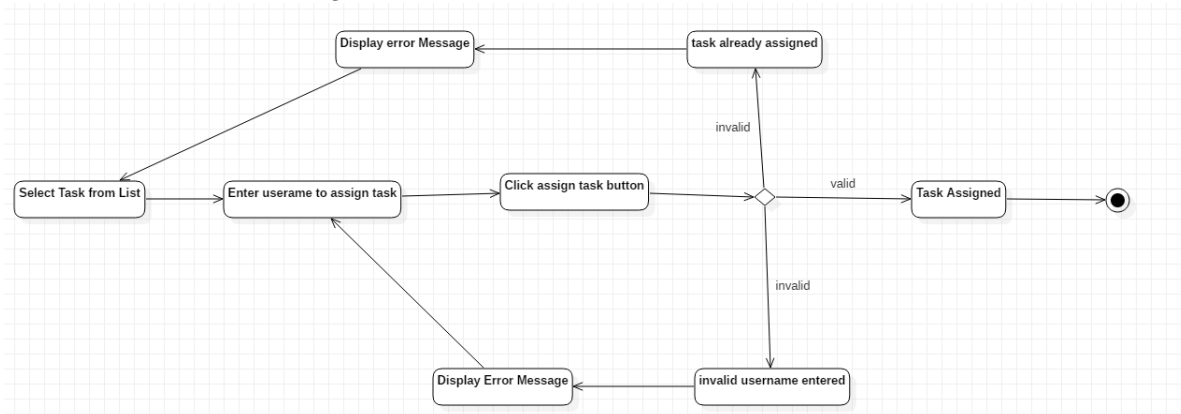
Return a motor to a customer state diagram

Inform customer of potential delay

delay is not okay — return motor to customer

delay is okay

Order Part

need another part

Part Arrives

| Motor arrives at garage | Technician will inspect motor | can be fixed | motor can be fixed | Technicians will work on motor | Motor Fixed | ● |

cannot be fixed

motor cannot be fixed

Labour time exceeds estimated

delay is okay

customer is informed of delay

delay is not okay

return motor to customer