|  |
| --- |
| **Use Case:** |
| **ID:** |
| **Brief description:** |
| **Primary actors:** |
| **Secondary actors:** |
| **Preconditions:** |
| **Main flow:** |
| **Postconditions:** |
| **Alternative flows:** |

addJob+

GenerateReport -

setCustomerAccount+

alterCustomerAccount +

allocateMechanic +

ProduceInvoice +

updateStock - Bao

logIn - Bao

logout - Bao

print - Bao

addUserAccount – Bao

|  |
| --- |
| **Use Case:** AddJob |
| **ID:** |
| **Brief description:**  Taking in new job and add them to the pending list of jobs. |
| **Primary actors:** Foreperson, Receptionist, Franchisee |
| **Secondary actors:**  Database |
| **Preconditions:**   1. System is operational 2. The user has successfully logged in the system |
| **Main flow:**   1. The use case starts when the user selects the addJob option in the system menu 2. The system prompts a form to the user. 3. User fills in the form and submits it to the system 4. The system establishes a connection with the database 5. The system copies the data to the database 6. The system disconnects from the database 7. The system informs the user that the job has been added to the list. |
| **Postconditions:**   1. The pending job list has been updated |
| **Alternative flows:**  connectionNotMade |

|  |
| --- |
| **Use Case:** AddJob:connectionNotMade |
| **ID:** |
| **Brief description:**  The system could not establish a connection to the database |
| **Primary actors:** Foreperson, Receptionist, Franchisee |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The user has logged in 3. The user has submitted the data for a new job |
| **Main flow:**   * 1. The use case starts at step 4 of the main flow when the system is unable to connect to the database to add the job to the pending job list.   2. The system informs the user that it could not connect to the database.   3. The user acknowledges the notification.   4. The system prompts the user with main menu |
| **Postconditions:**  None. |

|  |
| --- |
| **Use Case:** SetCustomerAccount |
| **ID:** |
| **Brief description:**  Adding the customer data to the database. |
| **Primary actors:** Franchisee |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The user has logged in successfully |
| **Main flow:**   1. The use case starts when the user selects the SetCustomerAccount option in the system menu 2. The system shows a form on the GUI. 3. User fills in the form and submits it to the system 4. The system establishes a connection with the database 5. The system copies the data to the database 6. The system disconnects from the database. 7. The system informs the user that the customer information has been added to the database |
| **Postconditions:**   1. The new customer data has been added to the database 2. The system successfully disconnected from the database after finishing adding the user |
| **Alternative flows:**  ConnectionNotMade |

|  |
| --- |
| **Use Case:** SetCustomerAccount:ConnectionNotMade |
| **ID:** |
| **Brief description:**  The system could not establish a connection to the database |
| **Primary actors:** Franchisee |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The user has logged in 3. The user has submitted the data for a new customer |
| **Main flow:**   * 1. The use case starts at step 4 of the main flow when the system is unable to connect to the database.   2. The system informs the user that it could not connect to the database.   3. The user acknowledges the notification.   4. The system prompts the user with main menu |
| **Postconditions:**  None. |

|  |
| --- |
| **Use Case:** AlterCustomerAccount |
| **ID:** |
| **Brief description:**  The franchisee can alter the customer accounts by searching for a specific customer and alter their data in the database. |
| **Primary actors:** Franchisee |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The user has logged in |
| **Main flow:**   1. The use case starts when the user selects AlterCustomerAccount from the main menu 2. The system connects to the database and gets all customer accounts 3. The customer account data is put in a java ArrayList 4. The system shows the user the list of customers on the GUI 5. The user highlights the account they want to alter and clicks the choose button 6. The system shows another form on the GUI with the customer account data 7. The user inputs the data they want to change 8. The user clicks the submit button 9. The System first deletes the customer they were changing in the database 10. The system saves the new data to the database. |
| **Postconditions:**   1. The new data has been successfully added to the database 2. The old data has been successfully deleted from the database |
| **Alternative flows:**  ConnectionNotMade |

|  |
| --- |
| **Use Case:**  AllocateMechanic |
| **ID:** |
| **Brief description:**  The foreperson can allocate a mechanic to a job from the pending job list, which is done by selecting a particular job from the pending list and then selecting the mechanic which will do the job. |
| **Primary actors:** Foreperson |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The Foreperson has successfully logged in the system 3. The pending job list is not empty |
| **Main flow:**   1. The use case starts when the user chooses AllocateMechanic option in the main menu. 2. The system connects to the database 3. The system retrieves the list of pending jobs. 4. The system puts the list of pending jobs in an ArrayList 5. The system shows the user the list of jobs. 6. The user highlights a specific job by clicking it in the list. 7. The user clicks select, hence chooses the job 8. The system retrieves the list of mechanics from the database 9. The system shows the user a list of mechanics. 10. The user highlights the specific mechanic they want to assign to the job. 11. The user clicks ok. 12. The system notifies the user that the mechanic has been successfully allocated to the job 13. The system disconnects from the database. |
| **Postconditions:**   1. The system has successfully changed the details of the job 2. The system successfully disconnected from the database |
| **Alternative flows:**  ConnectionNotMade |

|  |
| --- |
| **Use Case:** ProduceInvoice |
| **ID:** |
| **Brief description:**  The Receptionist should be able to produce an invoice for a recently finished job, which afterwards will be forwarded to the customer. This can be done by choosing the job the invoice should be done on from the GUI and getting a query to the database, which will fetch the information about the chosen job. |
| **Primary actors:** Receptionist, Foreperson |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The user is logged in the system 3. There are jobs finished |
| **Main flow:**   1. The use case starts when the user chooses the PendingJobList from the GUI 2. The system fetches the pending job list and copies the data to an ArrayList 3. The system displays the list on the GUI 4. The user highlights the job they want to produce an invoice for 5. The user clicks produce invoice button on the side 6. The system queries the database for each task done for this job 7. For each task:    1. The system copies the Task into a task object in the loop    2. The system appends the data of the task to the end of the invoice    3. The system adds the cost to session.total 8. The system calculates labour cost 9. The system appends the price to the end of the document 10. The system calculates the VAT 11. The system adds the VAT to the total and appends it to the end of the invoice 12. The system disconnects from database. |
| **Postconditions:**   1. The data in the invoice is correct 2. The system successfully disconnected from the database |
| **Alternative flows:**  ConnectionNotMade |

|  |
| --- |
| **Use Case:** GenerateReport |
| **ID:** |
| **Brief description:**  The system generates a report chosen by the user and shows it to the user in viewable and printable form. |
| **Primary actors:** Franchisee. |
| **Secondary actors:**  Database |
| **Preconditions:**   1. System is operational 2. The user has successfully logged in the system |
| **Main flow:**   1. The use case starts when the user selects Generate report from the main menu 2. The system shows a form on the GUI where the user can choose a specific type of report to generate 3. Connect to the Database 4. Execute database query with specified attributes 5. Format the query result 6. Present it to the user   Extension point: View Report, Print   1. Disconnect from Database |
| **Postconditions:**   1. Results successfully fetched 2. Successfully disconnected from the Database |
| **Alternative flows:**  **ConnectionNotMade**  **NoResult** |

|  |
| --- |
| **Use Case:** GenerateResport**:** ConnectionNotMade |
| **ID:** |
| **Brief description:**  The system failed to establish a connection to the system and database. |
| **Primary actors:** Franchisee, Receptionist |
| **Secondary actors:**  System, Database |
| **Preconditions:**   1. The system is operational 2. The user attempts to generate report, by filling in the required attributes |
| **Main flow:**   * 1. The use case starts at step 3 of the main flow when the system is unable to connect to the database to generate the report.      1. The system informs the user that it could not connect to the database.      2. The user acknowledges the notification.      3. The system prompts the user with main menu. |
| **Postconditions:**  None**.** |

|  |
| --- |
| **Use Case:** GenerateReport: NoResult |
| **ID:** |
| **Brief description:**  The system failed to generate a report, with the specified attributes |
| **Primary actor:**  Franchisee, Receptionist |
| **Secondary actors:**  System, Database |
| **Preconditions:**   1. The system is operational 2. Connection to the database is successful 3. The user attempts to generate report, giving some attributes (job type, etc.) |
| **Main flow:**   * 1. The use case starts at step 5 of the main flow when the system is unable to generate a report given the specified attributes      1. The system presents the user with a blank page, notifying the user that a report could not be specified given the context.      2. The user acknowledges the notification.      3. The system prompts the user to try again |
| **Postconditions:**  None**.** |

|  |
| --- |
| **Use Case:** logIn |
| **ID:** |
| **Brief description:**  The function allows the user to get access to the system & database by proving their own identity through an username and password that only that particular person knows. |
| **Primary actors:** Administrator, Franchisee, Foreperson, Mechanic, Receptionist |
| **Secondary actors:**  System, Database |
| **Preconditions:**   1. The system is operational |
| **Main flow:**   1. The use case started when the user selects the logIn function in the main menu. 2. The system prompts a form for the user. 3. User fills in the form and submits it to the system. 4. The system establishes a connection with the database. 5. The system matches the information on the form that the user filled in to the database. 6. If (while logIn exists):   6.1) The information on the form matches the information on the system about the user  6.2) The information on the form doesn’t match the information on the system about the user  6.2.1) System prompts a notification that notices the user that the information filled in is wrong and adjustment is required.  6.2.2) Loop back to 2).   1. The system prompts the user their profile page. |
| **Postconditions:**   1. User get access to their own profile page and get access to all the function that their role allowed them to perform on the system. |
| **Alternative flows:**  ConnectionNotMade  logInDetailsAreWrong |

|  |
| --- |
| **Use Case:** logIn**:**ConnectionNotMade |
| **ID:** |
| **Brief description:**  The system failed to establish a connection to the system and database. |
| **Primary actors:** Administrator, Franchisee, Foreperson, Mechanic, Receptionist |
| **Secondary actors:**  System, Database |
| **Preconditions:**   1. The system is operational 2. The user attempts to perform logIn function by filling the form and submits |
| **Main flow:**   * 1. The use case starts at step 6 of the main flow when the system is unable to connect to the database to allow the user to log in.      1. The system informs the user that it could not connect to the database.      2. The user acknowledges the notification.      3. The system prompts the user with main menu. |
| **Postconditions:**  None**.** |

|  |
| --- |
| **Use Case:** logIn:logIndetailsAreWrong |
| **ID:** |
| **Brief description:**  The system can’t match the information in the form filled by the user with the system to verify the user. |
| **Primary actors:** Administrator, Franchisee, Foreperson, Mechanic, Receptionist |
| **Secondary actors:**  System, Database |
| **Preconditions:**   1. The system is operational 2. The user attempts to perform logIn function by filling the form and submits. |
| **Main flow:**  6.2) The use case starts at step 6 of the main flow when the system is unable to match the information on the form to the information on the system about the user  6.2.1) System prompts a notification that notices the user that the information filled in is wrong and adjustment is required.  6.2.2) User acknowledges the notification  6.2.3) Loop back to step 2) in the main flow. |
| **Postconditions:**  None. |

|  |
| --- |
| **Use Case:** addUserAccount |
| **ID:** |
| **Brief description:**  Adding new user account for employees in the business branch. |
| **Primary actors:** Administrator |
| **Secondary actors:**  Database |
| **Preconditions:**   1. System is operational 2. The user has successfully logged in the system |
| **Main flow:**   1. The use case starts when the user selects the addUserAccount option in the system menu 2. The system prompts a form to the user. 3. User fills in the form and submits it to the system 4. The system establishes a connection with the database 5. The system copies the data to the database 6. The system disconnects from the database 7. The system informs the administrator that the new user has been added to the database. |
| **Postconditions:**   1. The user list in the database is updated with one new user. |
| **Alternative flows:**  connectionNotMade |

|  |
| --- |
| **Use Case:** addUserAccount:connectionNotMade |
| **ID:** |
| **Brief description:**  The system could not establish a connection to the database |
| **Primary actors:** Administrator |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The user has logged in 3. The user has submitted the data for a user account |
| **Main flow:**   * 1. The use case starts at step 4 of the main flow when the system is unable to connect to the database.   2. The system informs the user that it could not connect to the database.   3. The user acknowledges the notification.   4. The system prompts the user with main menu |
| **Postconditions:**  None. |

|  |
| --- |
| **Use Case:** logOut |
| **ID:** |
| **Brief description:**  The function allows user to remove their access to their account and their ability to perform functions on the system. |
| **Primary actors:** Administrator, Franchisee, Foreperson, Mechanics, Receptionist |
| **Secondary actors:**  System, Database |
| **Preconditions:**   1. System is operational 2. The user has successfully logged in the system |
| **Main flow:**   1. The use case started when the user performs the logOut function on the main menu. 2. The system establishes a connection to the system 3. The system removes the user account’s access to the database and the system 4. The system switches off the user account 5. The system disconnects from the database 6. The system prompts the main menu |
| **Postconditions:**   1. The user is switched off and get their access to the database removed. |
| **Alternative flows:**  connectionNotMade |

|  |
| --- |
| **Use Case:** logOut:connectionNotMade |
| **ID:** |
| **Brief description:**  The system could not establish a connection to the database |
| **Primary actors:** Administrator, Franchisee, Foreperson, Mechanics, Receptionist |
| **Secondary actors:**  System, Database |
| **Preconditions:**   1. The system is operational 2. The user has logged in 3. The user has attempted to perform logOut function |
| **Main flow:**   * 1. The use case starts at step 2 of the main flow when the system is unable to connect to the database.   2. The system informs the user that it could not connect to the database.   3. The user acknowledges the notification.   4. The system prompts the user back to their original destination |
| **Postconditions:**  None. |

|  |
| --- |
| **Use Case:** updateStock |
| **ID:** |
| **Brief description:**  This function allows the user to make connection to the database and readjust and update the stock level by manually inputting the stock data. |
| **Primary actors:** Receptionist |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The user is logged in |
| **Main flow:**   1. The use case started when the user performs the updateStock functions when stock notifications are on 2. The system prompts out a form for user to fill out the stock that they are updating and the amount 3. User fills in the forms and submit to the system 4. The system makes a connection to the database 5. The system copies the data to the database 6. The system disconnects from the database 7. The system then informs the user that the stock level in the database has been updated. |
| **Postconditions:**   1. The new order has been set 2. An estimated time to receive the order are being shown to the user |
| **Alternative flows:**  connectionNotMade |

|  |
| --- |
| **Use Case:** updateStock:connectionNotMade |
| **ID:** |
| **Brief description:**  The system could not establish a connection to the database |
| **Primary actors:** Receptionist |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The user has logged in 3. The user has attempted to perform updateStock function |
| **Main flow:**   * 1. The use case starts at step 4 of the main flow when the system is unable to connect to the database.   2. The system informs the user that it could not connect to the database.   3. The user acknowledges the notification.   4. The system prompts the user back to their original destination |
| **Postconditions:**  None. |

|  |
| --- |
| **Use Case: Print** |
| **ID:** |
| **Brief description:**  This function allows user to print out any type of documents such as reports, invoice, job sheets, etc… |
| **Primary actors:** Franchisee, Receptionist |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The user is logged in |
| **Main flow:**   1. The use case when user performs the function print function in the main menu 2. The system then prompts a list of possible documents that the user can print 3. The user then picks the document they want to perform the function 4. The system then confirms and show the documents that the user wants to print 5. The system then prompts a list of conditions that narrows down the specific conditions that user wants to print the document in 6. The user then selects the conditions they want from the list and confirm it. 7. The system makes a connection to the database 8. While (print exists):   7.1) System then arrange the final documents that has all the conditions gathered  7.2) System then prompts the final document for the user to ask for confirmation   1. The user then confirms the document that is shown to them 2. The user then performs the print function 3. The system then translates the document and do the printing function. |
| **Postconditions:**   1. The system successfully prints out the document in the correct format. |
| **Alternative flows:**  connectionNotMade  noDocumentsAccumulated  adjustmentNotMade |

|  |
| --- |
| **Use Case:** print:connectionNotMade |
| **ID:** |
| **Brief description:**  The system could not establish a connection to the database |
| **Primary actors:** Franchisee, Receptionist |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The user has logged in 3. The user has attempted to perform print function |
| **Main flow:**   * 1. The use case starts at step 6 of the main flow when the system is unable to connect to the database.   2. The system informs the user that it could not connect to the database.   3. The user acknowledges the notification.   4. The system prompts the user back to the condition list. |
| **Postconditions:**  None. |

|  |
| --- |
| **Use Case:** print:noDocumentsAccumulated |
| **ID:** |
| **Brief description:**  The system could not find the documents available in the database of the computer. |
| **Primary actors:** Franchisee, Receptionist |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The user has logged in 3. The user has attempted to perform print function |
| **Main flow:**   * 1. The use case starts at step 2) of the main flow when the system is unable to find the document available in the database   2. The system informs the user that it could not find the documents in the system.   3. The user acknowledges the notification.   4. The system prompts the user back to the main menu. |
| **Postconditions:**  None. |

|  |
| --- |
| **Use Case:** print:adjustmentNotMade |
| **ID:** |
| **Brief description:**  The system could not perform the printing function based on the printing settings that the user set up for the document to send to the printer. |
| **Primary actors:** Franchisee, Receptionist |
| **Secondary actors:**  Database |
| **Preconditions:**   1. The system is operational 2. The user has logged in 3. The user has attempted to perform print function |
| **Main flow:**   * + 1. The use case starts at step 8.1) of the main flow when the system is unable to perform the print function based on the options to print that is being sent by the user     2. The system informs the user that it could not print based on the print option list that the user sent.     3. The user acknowledges the notification.     4. The system prompts the user back to the print option list. |
| **Postconditions:**  None. |