

|  |  |
| --- | --- |
| Use Case | See what tasks there are |
| Actor(s) | Technician, Database Operator |
| Primary Flow | 1. Technician opens the application using the shared computer. 2. Technician finds the order they are interested in among the list of **all orders.** 3. Technician press’s “View” button under “problems” column 4. The window displaying all the problems connected to this order is displayed. |
| Alternative Flow | 1. Database Operator finds the database file. 2. Database Operator opens the database file storing all currently existing problems and the orders they are connected to. |

|  |  |
| --- | --- |
| Use Case | See all orders |
| Actor(s) | Technician |
| Primary Flow | 1. Technician opens the application using the shared computer 2. Technician finds the order they are interested in among the list of all orders. |
| Alternative Flow | 1. Database Operator finds the database file 2. Database Operator opens the database file storing all currently existing orders. |

|  |  |
| --- | --- |
| Use Case | Assign technicians to the task |
| Actor(s) | Technician, Database Operator |
| Primary Flow | 1. Technician opens the task allocation application. 2. Technician finds the problem they want to change assignment at. 3. Technician selects the problem. 4. Technician uses “change technician” button with selected problem. 5. From the pop-up window, the technician chooses other technician. 6. With selected technician, user clicks on “Change Technician”, confiriming the choice. 7. Technician informs the new technician, about the change. |
|  | 1. Database Operator finds the database with problems. 2. Database Operator finds the problem they want to change assignment to. 3. Database Operator changes the “Technician” value at relevant problem to the technician’s name. 4. Database Operator uploads the changed database to the server. |

|  |  |
| --- | --- |
| Use Case | Add problems |
| Actor(s) | Technicians |
| Primary Flow | 1. Technician opens application using shared computer. 2. At the main screen, technician find the order they want to add problem to. 3. Technician opens problems for that order by pressing “View” button. 4. Technician uses “Add problem” button and writes down specifics. 5. Technician confirms changes pressing “add problem”. 6. System updates the database. |

|  |  |
| --- | --- |
| Use Case | Complete problems |
| Actor(s) | Technicians, Database Operator |
| Primary Flow | 1. Technician completes the task. 2. Technician opens the application 3. At the main screen, technician finds the order they want to complete a problem from. 4. Technician opens problems for that order by pressing “View” button. 5. Technician selects the problem they have completed. 6. Technician uses “Delete problem” button below. |
| Alternative Flow | 1. User opens the application. 2. User finds the order they want to delete problem from. 3. User uses “Delete Order” button, to complete order and all problems in it. |
| Exception | 1. User wants to complete the problem they have just completed. 2. User opens the application. 3. User finds the order they want to delete order from. 4. Due to mismanagement, the order has no problems pending, or there is no problem the user was looking for. 5. User informs the rest of technicians and manager about the issue. |

|  |  |
| --- | --- |
| Use Case | See deadlines |
| Actor(s) | Technician |
| Primary Flow | 1. Technician opens the application using the shared computer 2. Technician finds the order they are interested in among the list of all orders. 3. Technician looks up for the deadline of the order they are looking for |
| Alternative Flow | 1. Database Operator finds the database file 2. Database Operator opens the database file storing all currently existing orders. 3. Database Operator find the order they are looking for and looks up its final inspection date column |

|  |  |
| --- | --- |
| Use Case | Add orders |
| Actor(s) | Technicians, Database Operator |
| Primary Flow | 1. User opens the main screen of the application. 2. User presses “Add Motor” button. 3. User enters all relevant information to the window, and presses “Add Motor” |
| Alternative Flow | 1. Database Operator finds the database file with orders. 2. Database Operator opens the file. 3. Database Operator adds an order row into the database and fills all relevant information 4. Database Operator uploads the database to the server. |

|  |  |
| --- | --- |
| Use Case | Add technicians |
| Actor(s) | Database Operator |
| Primary Flow | 1. Database Operator finds the database with technicians. 2. Database Operator adds a row to the database. 3. They fill the row with all relevant information about the technician. 4. Database Operator uploads the database to the server. |