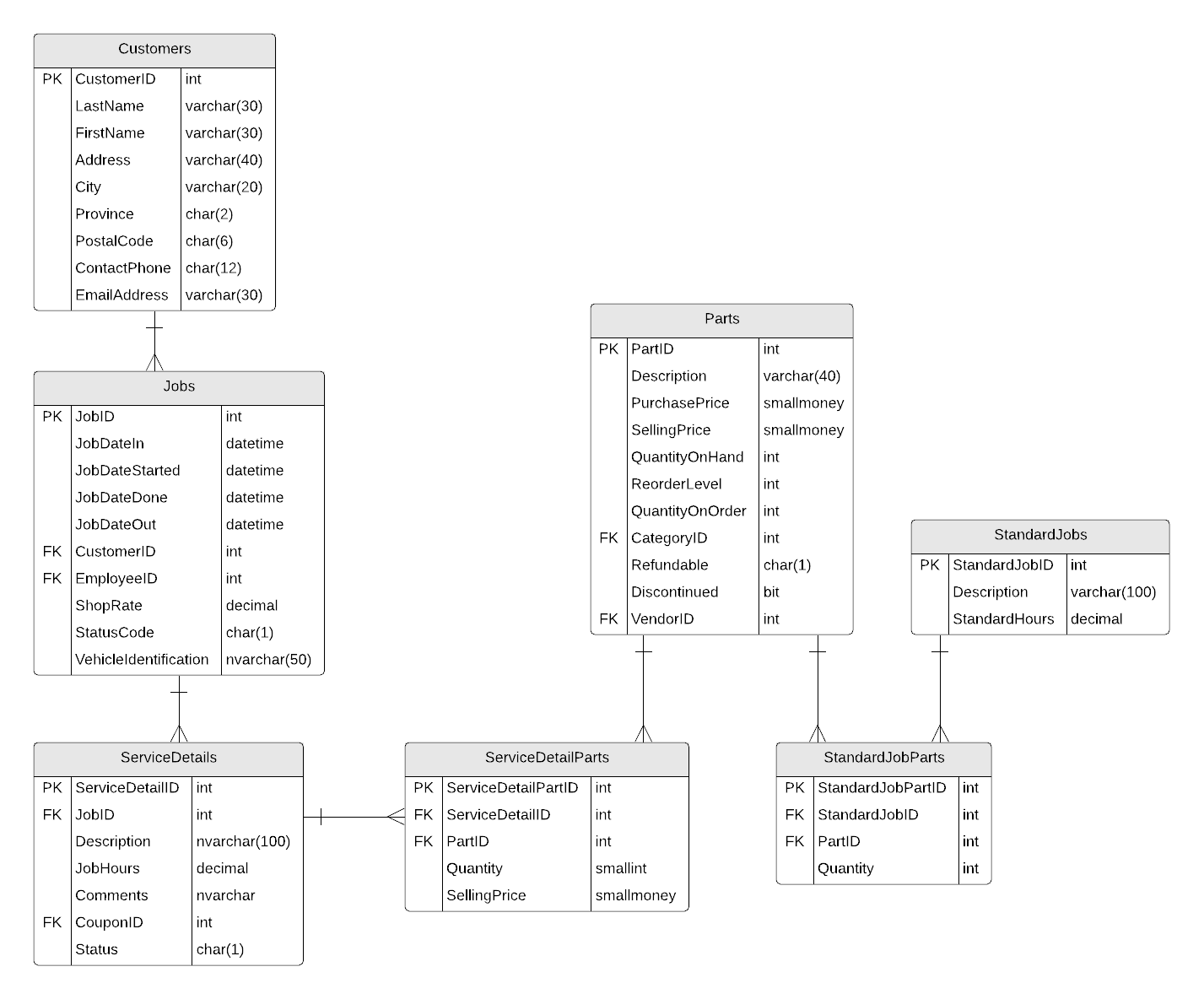
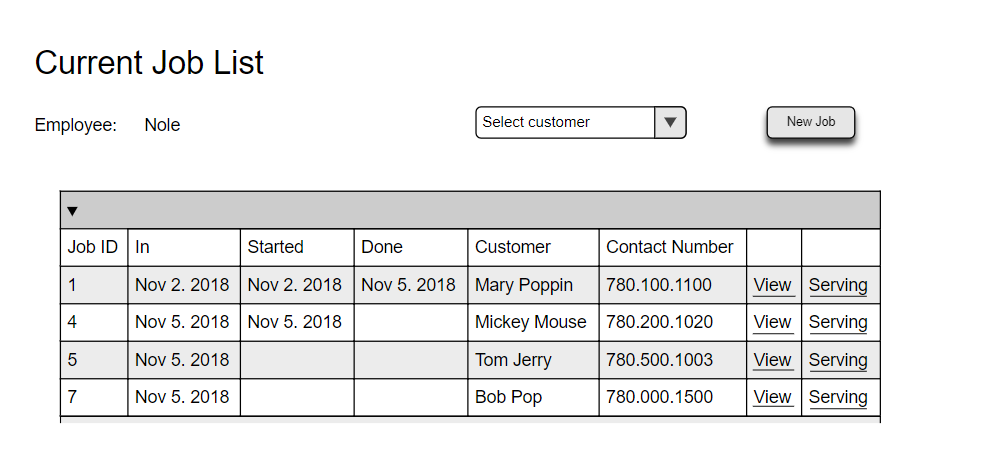
Jobing Subsystem

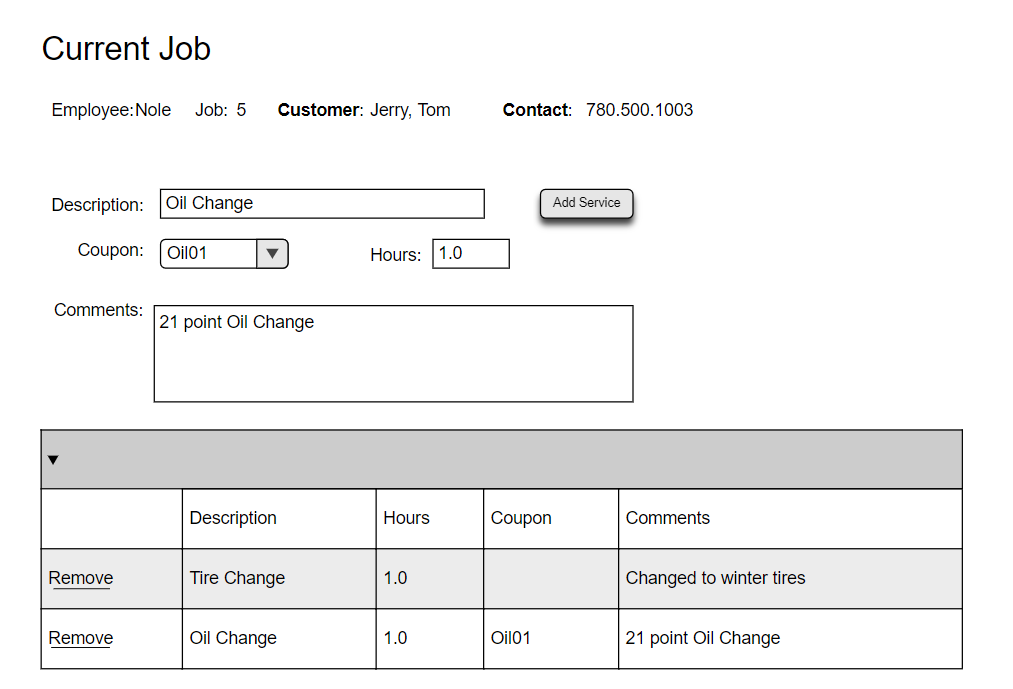
# ERD for UX process

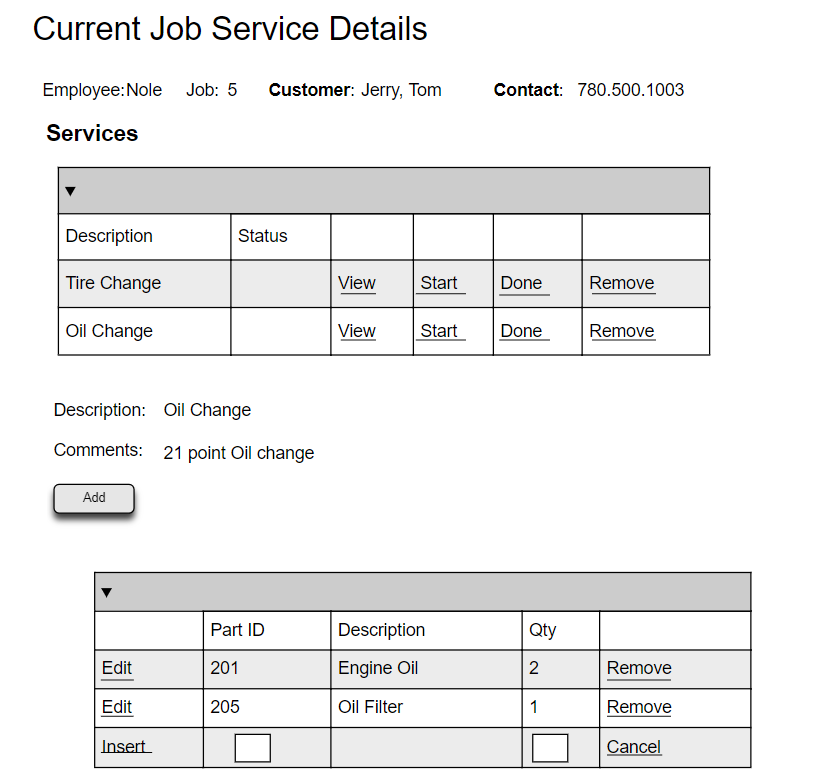
ERD diagram

# Possible UX Interface

Mock Up Screenshots







# Event and Wiring Summations

**JOBING - CURRENT JOBS**

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| CustomerDDL | | Wired via ODS |
| **BLL Class(es) and Method (s)** | | |
| CustomerController | List <CustomerList> Customer\_CustomerList()   * Generate the customer List (Customer ID, Name Display) | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| Customers(R) | | CustomerList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| CurrentJobList | | Wired via ODS |
| **BLL Class(es) and Method (s)** | | |
| JobController | List <JobList> Job\_CurrentJobList()   * Generate the customer List (Customer ID, Name Display) | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| Customers(R) Job(R) | | JobList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| NewJob | | NewJob OnClick   * Validate data present – customer is selected; * Error message: If no value is selected * Parse Customer information to the new job page |
| **BLL Class(es) and Method (s)** | | |
|  |  | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
|  | |  |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| View | | View OnClick   * Parse the Job ID which will contain; * Customer Information * Existing service information |
| **BLL Class(es) and Method (s)** | | |
|  |  | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
|  | |  |

**VIEW/ EDIT CURRENT JOBS**

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| StandardJobListing | | * Wired via ODS |
| **BLL Class(es) and Method (s)** | | |
| StandardJobController | List < StandardJobList > Job\_ StandardJobListing ()   * Generate the standard Job List (description, hours)   Void Add\_StandardJob (string description, decimal standardhours)   * Create a new standard job service | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| StandardJob(C, R) | | StandardJobList(P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| CurrentJobService | | Wired via ODS |
| **BLL Class(es) and Method (s)** | | |
| JobServiceController | List<JobServiceList> List\_ CurrentJobService (int jobid)   * Retrieve a list of job services for supplied jobid * Return description, hours, coupons and comments | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| ServiceDetails(R) Job(R) | | JobServiceList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| AddJobService, Add Service | | OnClick / Wired via ODS |
| **BLL Class(es) and Method (s)** | | |
| JobServiceController | Void Add\_JobServiceList (int jobid)   * add a job service * add description, hours, coupons and comments * bind data with Service Details | |
| **SQL Table(s): (C, R, U, D)** | | **Entities/DTOs/POCOs** |
| ServiceDetails(C,R) Job(R) | | JobServiceList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Remove | | Remove.Onclick/ Code behind   * remove information about job service * Error message: if job service has been removed by another user * bind data with Service Details * Refresh page |
| **BLL Class(es) and Method (s)** | | |
| JobServiceController | Void Remove\_JobServiceList (int jobid) | |
| **SQL Table(s): (C, R, U, D)** | | **Entities/DTOs/POCOs** |
| ServiceDetails(R,D) Job(R) | | JobServiceList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| DisplayCoupons | | Code behind   * Check valid coupon information for service (Startdate, enddate, coupondiscount) * Error message: if one or more information is incorrect or if no available coupon * bind data with Service Details * Refresh page |
| **BLL Class(es) and Method (s)** | | |
| PopulateCoupons | Add Check\_JobServiceList (datetime startdate, datetime enddate, int coupondiscount)   * Add coupon discount to the service details | |
| **SQL Table(s): (C, R, U, D)** | | **Entities/DTOs/POCOs** |
| ServiceDetails(R,U) Coupons(C) | | JobServiceList (P) |

**VIEW/ EDIT CURRENT JOB SERVICE**

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Serving | | Serving OnClick   * Parse Customer information to the current job page * Parse existing service information * Parse details of the current service detail being worked on |
| **BLL Class(es) and Method (s)** | | |
| JobServiceController | List < JobServiceList > Job\_CurrentJobServiceList(jobid)   * Generate the customer service details (job id, description) | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| ServiceDetails(R) Job(R) Customer(R) | | JobServiceList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| CurrentServicePartList | | Wired via ODS |
| **BLL Class(es) and Method (s)** | | |
| ServicePartController | List < ServicePartList > ServicePart\_CurrentServicePartList()   * Generate the service part details (part id, description, qty) | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| ServicePartDetails(R) ServiceDetails(R) Job(R) Customer(R) | | ServicePartList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| Edit | | OnClick/Code behind   * edit a job service * bind data * refresh page |
| **BLL Class(es) and Method (s)** | | |
| ServicePartController | Void Edit\_ServicePartList (int partid, int servicedetailid,) | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| ServicePartDetails(U) | | ServicePartList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| RemovePart | | OnClick /Code behind   * Delete service part if unused * Update Quantity on Parts table * bind data * refresh page |
| **BLL Class(es) and Method (s)** | | |
| ServicePartController | Void Remove\_ ServicePartList (int partid, int servicedetailid,) | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| ServicePartDetails(D) | | ServicePartList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| RemoveService | | OnClick/Code behind   * Check if job has been started * If yes: send an error message * If no (that is status is empty): then remove service * bind data * refresh page |
| **BLL Class(es) and Method (s)** | | |
| JobServiceController | Void Remove\_ JobServiceList (int partid, int servicedetailid,) | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| ServiceDetails(D) | | JobServiceList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| AddPart | | OnClick /Code behind   * Check all fields have been filled * Insert the new part to ServicePartDetails * Update Quantity on Parts table * bind data * refresh page |
| **BLL Class(es) and Method (s)** | | |
| ServicePartController | Void Add\_ ServicePartList (int partid, int servicedetailid,) | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
| ServicePartDetails(D) | | ServicePartList (P) |

|  |  |  |
| --- | --- | --- |
| **Controls** | | **Events** |
| ServicePartStatus | | OnClick /Code behind   * If status is; * empty: service has not started * start: service has started * done: service has been completed * status is updated when the Start/Done button is clicked * bind data * refresh page |
| **BLL Class(es) and Method (s)** | | |
|  |  | |
| **SQL Table(s): (C,R,U,D)** | | **Entities/DTOs/POCOs** |
|  | |  |