1 Planteamiento general del problema

Necesitamos una solución automatizada que ayude a gestionar las actividades de un taller mecánico. Este software, al que llamaremos CWS, deberá gestionar los vehículos de los clientes, las piezas de repuesto, las órdenes de trabajo, la formación de sus empleados, etc.

2 Casos de uso

2.1 Gestión de vehículos (Vehicle Management)

Cuando un cliente trae un vehículo al taller por primera vez, **el jefe del taller (foreman) debe dar de alta el vehículo en el sistema**. Si también es la primera vez del cliente, el jefe de taller deberá dar de alta también al cliente.

2.2 Gestión de trabajos (Work Scheduler)

El jefe del taller inspeccionará el vehículo, creará una nueva **orden de trabajo (work order) que detalle los trabajos a realizar y asignará el trabajo a un mecánico**.

El mecánico comprueba qué órdenes de trabajo tiene asignadas. Más tarde, inspecciona el vehículo en profundidad, decide las intervenciones (interventions) a realizar y las lleva a cabo.

Una vez realizadas todas las intervenciones, el **mecánico marca la orden de trabajo como FINISHED**, anota una descripción general del trabajo realizado y, para cada intervención, **registra el tiempo que empleó y los repuestos utilizados**, si los hubiera.

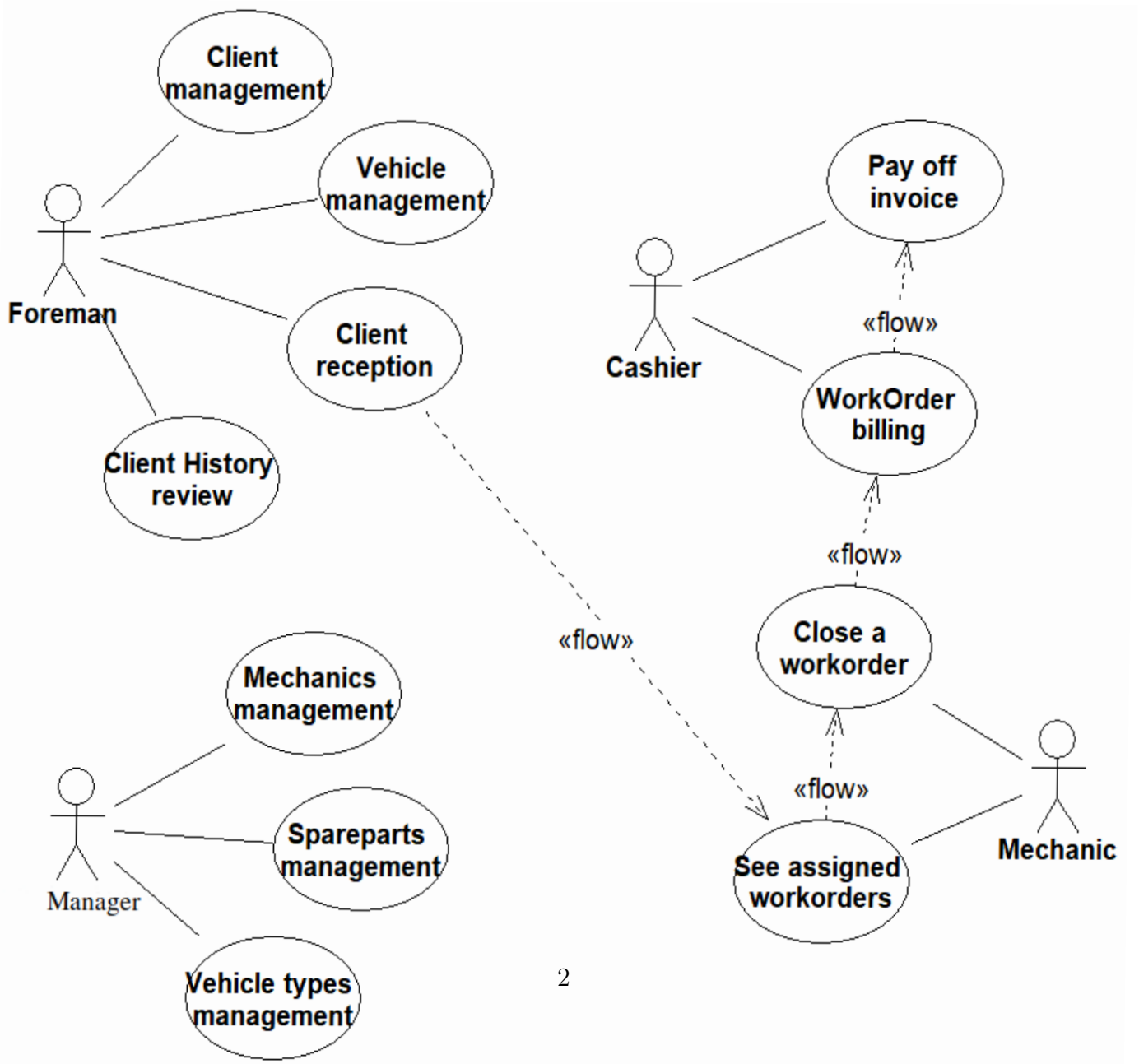
2.3 Gestión de facturas (Invoice Management)

En algún momento después de que el mecánico marque la orden de trabajo como terminada, **el cajero (cashier) debe crear una factura a partir de la orden de trabajo, o de varias. Todas ellas deben estar terminadas y no facturadas**.

Cuando el cliente acude al taller a recoger su vehículo, deberá abonar la factura. Los **métodos de pago aceptados** incluyen **efectivo** (cash), **tarjetas de crédito** y **vales de descuento emitidos por el taller (vouchers)**.

Jefe de taller Foreman	• Gestión de clientes • Gestión de vehículos • Recepción en taller • Revisar historial de un cliente
Mechanic	• Ver trabajos asignados • Marcar trabajo como finalizado
Cajero Cashier	• Facturar trabajo • Liquidar factura
Manager Gerente	• Gestión de mecánicos • Gestión de repuestos • Gestión de tipos de vehículo

2.4 Actores del sistema y casos de uso



3 Descripción de los casos de uso

3.1 Jefe de taller. Recepción en taller, check in

Cada vez que un vehículo venga a nuestro taller por primera vez, el jefe de taller (foreman) lo da de alta en el sistema (ver "Gestión de vehículos").

Luego, independientemente de que sea un vehículo nuevo o no, el jefe de taller inspeccionará el coche y **creará una orden de trabajo**, identificando al vehículo por su matrícula.

Una orden de trabajo recién creada estará en **estado OPEN**. Cuando se asigna a un mecánico pasará a **estado ASSIGNED**.

El jefe de taller podrá necesitar realizar, adicionalmente, las siguientes acciones:

- Modificar datos de la orden de trabajo
- Eliminar la orden de trabajo
- Ver historial de órdenes de trabajo del vehículo
- Ver vehículos del cliente
- Ver detalle de una orden de trabajo
- Buscar un vehículo por NIF del cliente, por matrícula o por marca y modelo.

3.2 Jefe de taller. Gestión de vehículos

La primera vez que un vehículo venga al taller, **el jefe de taller (foreman) lo da de alta en el sistema**. Debe introducir la matrícula, marca (brand), modelo (model), tipo de vehículo y propietario (owner). Si el propietario fuese un cliente nuevo, debería registrarlo previamente (ver *"Gestión de clientes"*).

Estos datos podrán ser actualizados cuando se necesite **pero** un vehículo únicamente podrá ser eliminado del sistema si no hay registrada en el sistema ninguna orden de trabajo (en ningún estado).

3.3 Jefe de taller. Gestión de clientes

Se deben realizar las operaciones CRUD básicas para los clientes.

El jefe de taller dará de alta un cliente la primera vez que acuda al taller. Se guardará su NIF, nombre, apellidos, dirección postal, teléfono y correo electrónico. Más tarde, podrá actualizar la información si fuese necesario e incluso eliminar al cliente.

Sin embargo, un cliente no podrá ser eliminado si tiene vehículos registrados a su nombre.

3.4 Jefe de taller. Revisar historial de un cliente

El jefe de taller podrá ver todos los vehículos registrados a nombre de un cliente así como todas las órdenes de trabajo registradas para cada vehículo.

También podrá buscar clientes en el sistema utilizando datos incompletos o a través de sus vehículos.

3.5 Mecánico. Listado de órdenes de trabajo asignadas

Cada mecánico podrá ver las órdenes de trabajo asignadas que tenga pendientes, actualizar el estado de la misma a FINISHED o ver el detalle de una orden de trabajo concreta, incluyendo las intervenciones hechas y los repuestos utilizados.

También podrá revisar el historial del vehículo.

Una orden de trabajo puede haber sido asignada varias veces, sucesivamente, a varios mecánicos.

3.6 Mecánico. Gestión de órdenes de trabajo y repuestos

Cada orden de trabajo puede suponer una o varias intervenciones en el vehículo. Para cada intervención, el mecánico debe registrar el tiempo utilizado (labour time) y la cantidad y tipo de repuestos (spare parts) empleados. El mecánico tendrá opciones para buscar repuestos por descripción, marca, modelo o código.

Una vez el mecánico ha completado la orden de trabajo, la marcará como FINISHED.

3.7 Cajero. Facturar una o varias órdenes de trabajo

La persona responsable de la caja será la encargada de **generar facturas**. Se podrán **facturar varias órdenes de trabajo en la misma factura**.

Para hacerlo, el cajero busca las **órdenes de trabajo no facturadas a partir del NIF** del cliente y, a continuación, genera una factura que las incluya y las marca todas como **INVOICED** (facturadas).

La generación de la factura debe ajustarse a ciertas restricciones:

- Sólo se podrán incluir órdenes de trabajo cuando sean del mismo cliente y su estado sea **FINISHED**.
- Varias órdenes de trabajo terminadas y no facturadas pueden cargarse en la misma factura. Para hacer esto, se le pide al cajero que introduzca los identificadores de las órdenes de trabajo que se cargarán en esta factura y se calcula el importe de cada una.
- El coste de una orden de trabajo se calcula como la **suma del importe de todas las intervenciones realizadas** que, a su vez, será **la suma del importe de mano de obra y de repuestos**.
  - El importe de mano de obra de una intervención depende del tiempo empleado (se guarda en minutos) y del precio de la hora de mecánico para ese tipo de vehículo. Aunque el precio se fije por horas, el cálculo se hace por minutos.
  - El importe de repuestos resulta de sumar el importe de todas las sustituciones. Y éste, a su vez, resulta de multiplicar el precio por unidad del repuesto por la cantidad empleada.
- Cuando el coste de todas las órdenes de trabajo a facturar está calculado se crea una factura.
  - La nueva factura se crea en el estado **NOT_YET_PAID** y las órdenes de trabajo facturadas pasan a estado **INVOICED**.
  - El **coste total de la factura** se calcula sumando el coste individual de todas las órdenes de trabajo facturadas **más impuestos (VAT)**. El VAT depende de la fecha en la que se haya generado la factura: si es anterior al 1/7/2012 se aplicará el 18%, a partir de entonces el 21%.
  - El importe total de la factura debe redondearse a 2 decimales.
  - La fecha de la factura será la del día en que se genera, sin precisión de hora.
  - Toda factura está identificada por un número de factura, que, por razones legales (Ministerio de Hacienda), debe ser único, secuencial y sin saltos (no puede haber la factura de nº 1012 si no hay la 1011).

3.8 Cajero. Liquidación de factura

Antes de llevarse el vehículo, el cliente debe pasar por caja para **abonar la factura**. El encargado de la caja debe buscar la factura por número de factura.

El sistema le mostrará los **medios de pago disponibles para ese cliente**. Podrán ser metálico, tarjeta de crédito o bonos. **Puede fraccionarse el pago** de una factura entre todos los medios

disponibles para ese cliente. Cada pago hecho para cubrir el total de una factura es un **cargo (charge)**.

Se debe considerar:

- El pago en metálico siempre está disponible para todo cliente.
- Si el cliente tiene tarjetas registradas en el sistema, podrá emplearlas si no estás caducadas.
- Si el cliente tiene uno o varios bono(s) (voucher), podrá pagar con él(ellos) sin pasarse de la cantidad disponible en el bono.
- No se pueden añadir varios cargos al medio de pago efectivo, la misma tarjeta o el mismo bono en la misma factura aunque sí se podrían utilizar varias tarjetas o bonos.
- El sistema llevará cuenta, para cada medio de pago (efectivo, tarjeta o bono descuento), del total pagado (acumulado) hasta la fecha.

Una vez que el importe total de la factura ha sido satisfecho, con uno o varios cargos en los medios de pago del cliente, la factura pasará al estado **PAID (abonada)**.

3.9 Manager (Gerente). Gestión de mecánicos

El manager (gerente del taller) será el encargado de gestionar las altas, bajas y modificaciones de los mecánicos. La baja de mecánicos es una operación inusual, que sólo será posible si en el sistema no hay más información acerca del mecánico que la información personal (no tiene reparaciones ni intervenciones).

El gerente también podrá listar los mecánicos registrados en el sistema.

3.10 Manager. Gestión de Repuestos

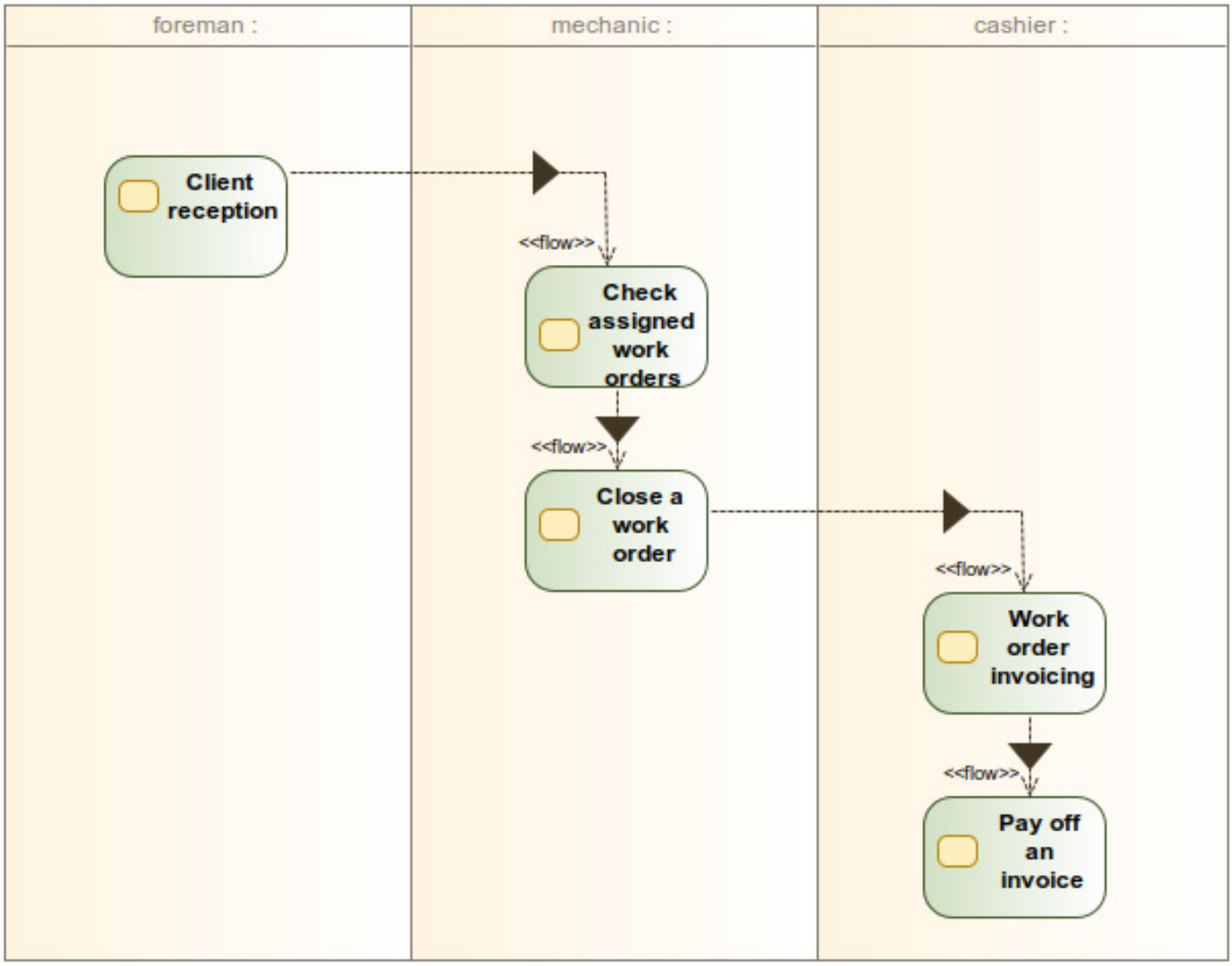
También administrará el inventario de repuestos. Puede agregar, actualizar y eliminar repuestos. También puede consultar repuestos por código, marca, modelo o descripción. El resultado de una búsqueda mostrará cuántas unidades se han utilizado de ese repuesto y el importe total ingresado por cada uno.

La modificación de los precios afectará a todas las facturas que se generen con posterioridad.

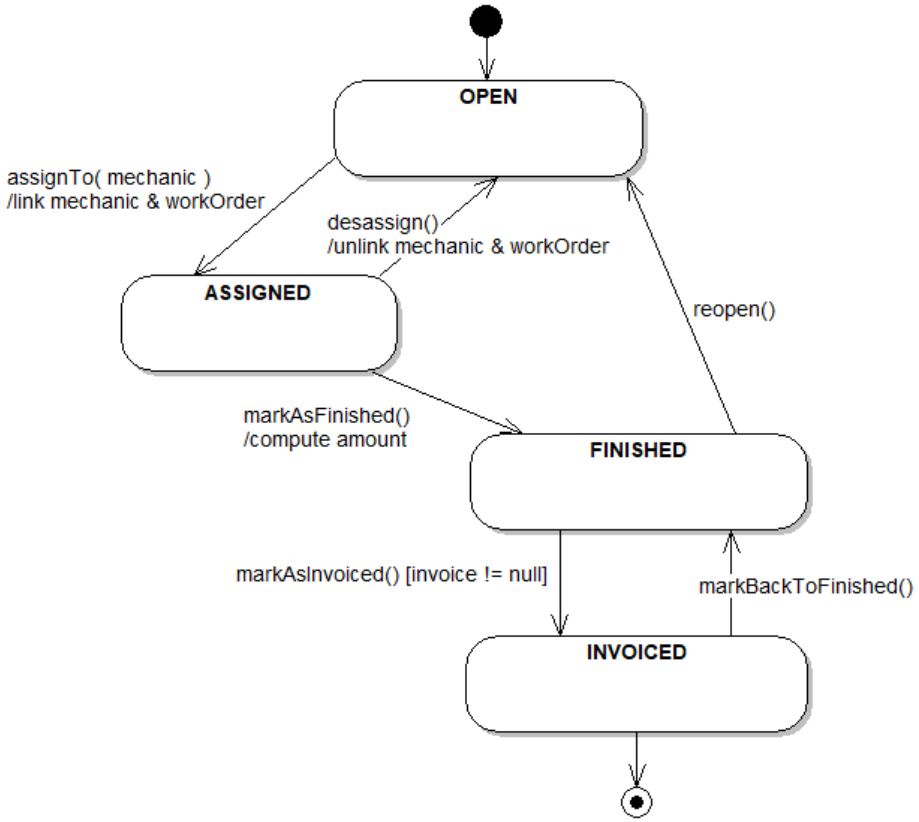
3.11 Manager. Gestión de tipos de vehículo

Respecto a los tipos de vehículos, podrá dar altas, bajas y modificaciones de los tipos de vehículo que se mantienen en el taller. La modificación más importante será la actualización del precio por hora de mecánico (mechanic labour rate) de cada tipo de vehículo. La modificación de los precios afectará a todas las facturas que se produzcan con posterioridad.

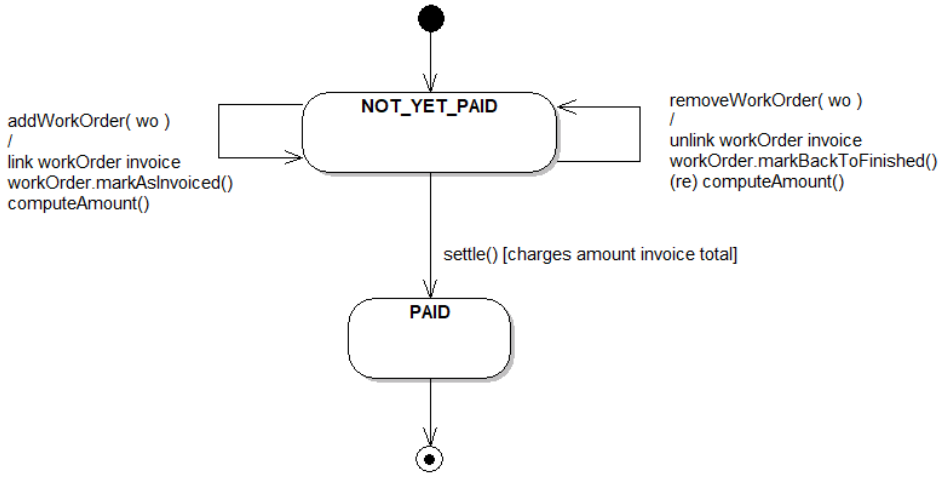
4 Workflow de casos de uso para una orden de trabajo



4.1 Diagrama de estados de una orden de trabajo



4.2 Diagrama de estados de una factura



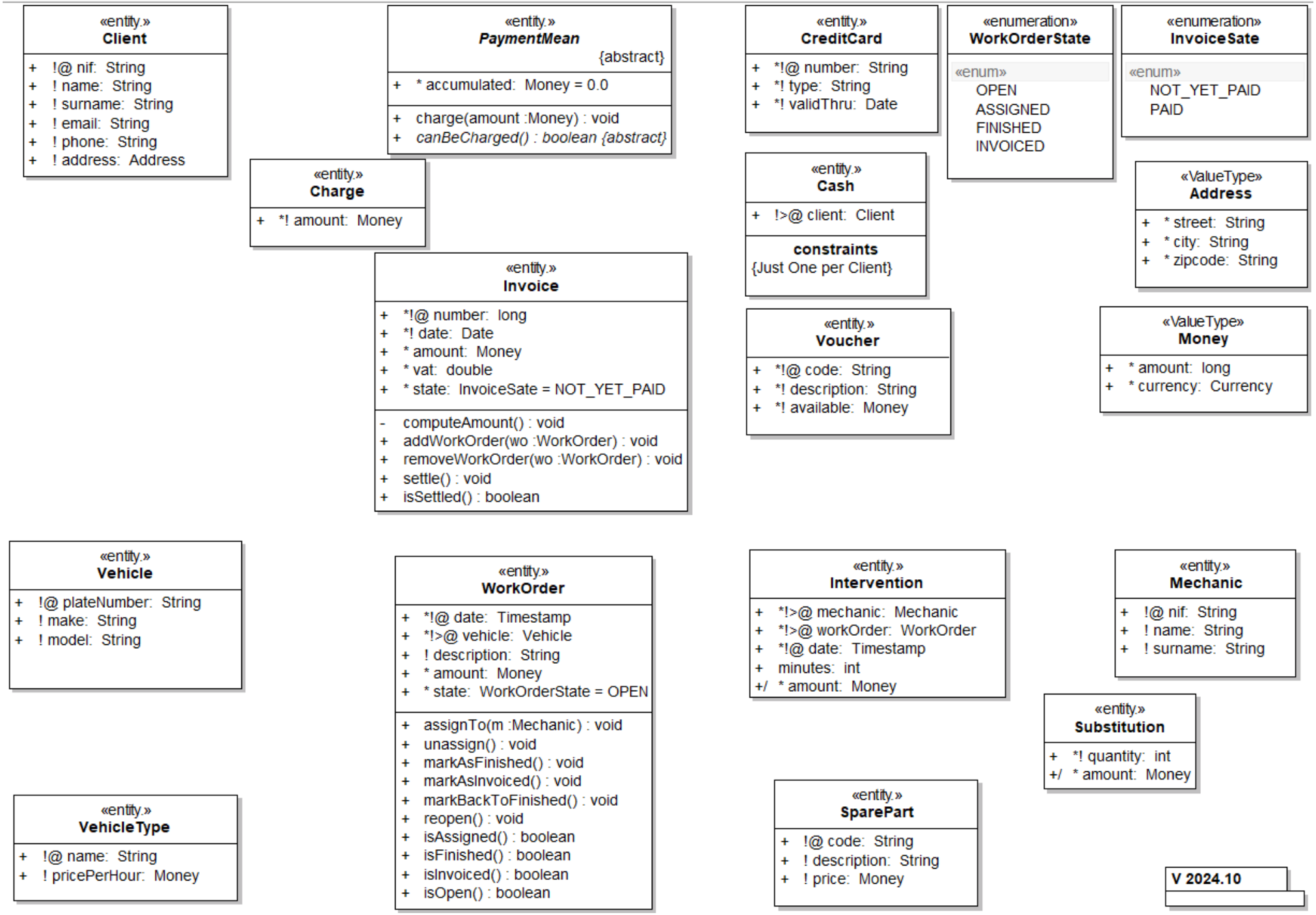
5 Completar modelo del dominio (PENDIENTE)

En la siguiente sesión se proporcionará la solución al modelo de dominio. Se han definido algunos símbolos adicionales al UML estándar que facilitan la elaboración del diagrama y que puedes utilizar también si lo deseas:

*	public readonly (getter only)
+	public usual (getter y setter)
-	private (no getter, no setter)
/	calculado, virtual (getter only)
!	inicializado en el full constructor
@	parte de la identidad natural de la entidad
>	atributo que forma parte de una asociación (atributo accidental)

Además de eso,

- las entidades asociativas (Associative-class entities) tienen como identidad natural las identidades naturales de ambos extremos de la asociación y se inicializan siempre en el full constructor.
- Los campos de tipo enum se inicializan siempre al primer valor de la enumeración
- Los campos Money se inicializan a EUR 0


«entity.»
Client
+ !@ nif: String
+ ! name: String
+ ! surname: String
+ ! email: String
+ ! phone: String
+ ! address: Address
«entity.»
PaymentMean
{abstract}
+ * accumulated: Money = 0.0
+ charge(amount :Money) : void
+ canBeCharged() : boolean {abstract}
«entity.»
CreditCard
+ *!@ number: String
+ *! type: String
+ *! validThru: Date
«enumeration»
WorkOrderState
«enum»
OPEN
ASSIGNED
FINISHED
INVOICED
«enumeration»
InvoiceSate
«enum»
NOT_YET_PAID
PAID
«entity.»
Charge
+ *! amount: Money
«entity.»
Cash
+ !>@ client: Client
constraints
{Just One per Client}
«ValueType»
Address
+ * street: String
+ * city: String
+ * zipcode: String
«entity.»
Invoice
+ *!@ number: long
+ *! date: Date
+ * amount: Money
+ * vat: double
+ * state: InvoiceSate = NOT_YET_PAID
- computeAmount() : void
+ addWorkOrder(wo :WorkOrder) : void
+ removeWorkOrder(wo :WorkOrder) : void
+ settle() : void
+ isSettled() : boolean
«entity.»
Voucher
+ *!@ code: String
+ *! description: String
+ *! available: Money
«ValueType»
Money
+ * amount: long
+ * currency: Currency
«entity.»
Vehicle
+ !@ plateNumber: String
+ ! make: String
+ ! model: String
«entity.»
WorkOrder
+ *!@ date: Timestamp
+ *!>@ vehicle: Vehicle
+ ! description: String
+ * amount: Money
+ * state: WorkOrderState = OPEN
+ assignTo(m :Mechanic) : void
+ unassign() : void
+ markAsFinished() : void
+ markAsInvoiced() : void
+ markBackToFinished() : void
+ reopen() : void
+ isAssigned() : boolean
+ isFinished() : boolean
+ isInvoiced() : boolean
+ isOpen() : boolean
«entity.»
Intervention
+ *!>@ mechanic: Mechanic
+ *!>@ workOrder: WorkOrder
+ *!@ date: Timestamp
+ minutes: int
+/ * amount: Money
«entity.»
Mechanic
+ !@ nif: String
+ ! name: String
+ ! surname: String
«entity.»
Substitution
+ *! quantity: int
+/ * amount: Money
«entity.»
VehicleType
+ !@ name: String
+ ! pricePerHour: Money
«entity.»
SparePart
+ !@ code: String
+ ! description: String
+ ! price: Money
V 2024.10