
CSCU9P5
Group Project
Group 01

Members:

2610846

2636568

2636159

2618009

2623637

Contents Page

Part	Name	Page No.
Part 1		
	a) Assumptions	3
	b) Use Case Diagrams	4 - 15
Part 2		
	a) Static Design Model	
	Class Diagram	16
	Purpose and Functionality Descriptions	17 - 18
	b) Dynamic Design Model	
	Sequence Diagrams	19 - 20
	State Diagrams	21 - 24

Part 1

A) Assumptions

Customer Relations Staff will be known as Reception. There are many Reception Staff using the same system. Reception's role is to take Clients phone calls and create the tickets. This means that Clients have no direct interaction with the system, all of the Client interaction is done through Reception staff. When a ticket is made by the receptionist the technician is notified of the job they have been assigned automatically through the mobile system used by all technicians.

The receptionist has the ability to call the client at anytime to collect or relay any further information such as, in our case, a new contract.

The car type is assumed to refer to the style of car that the manufacturer assigned e.g. hatchback, saloon, SUV etc. rather than the make or model of car.

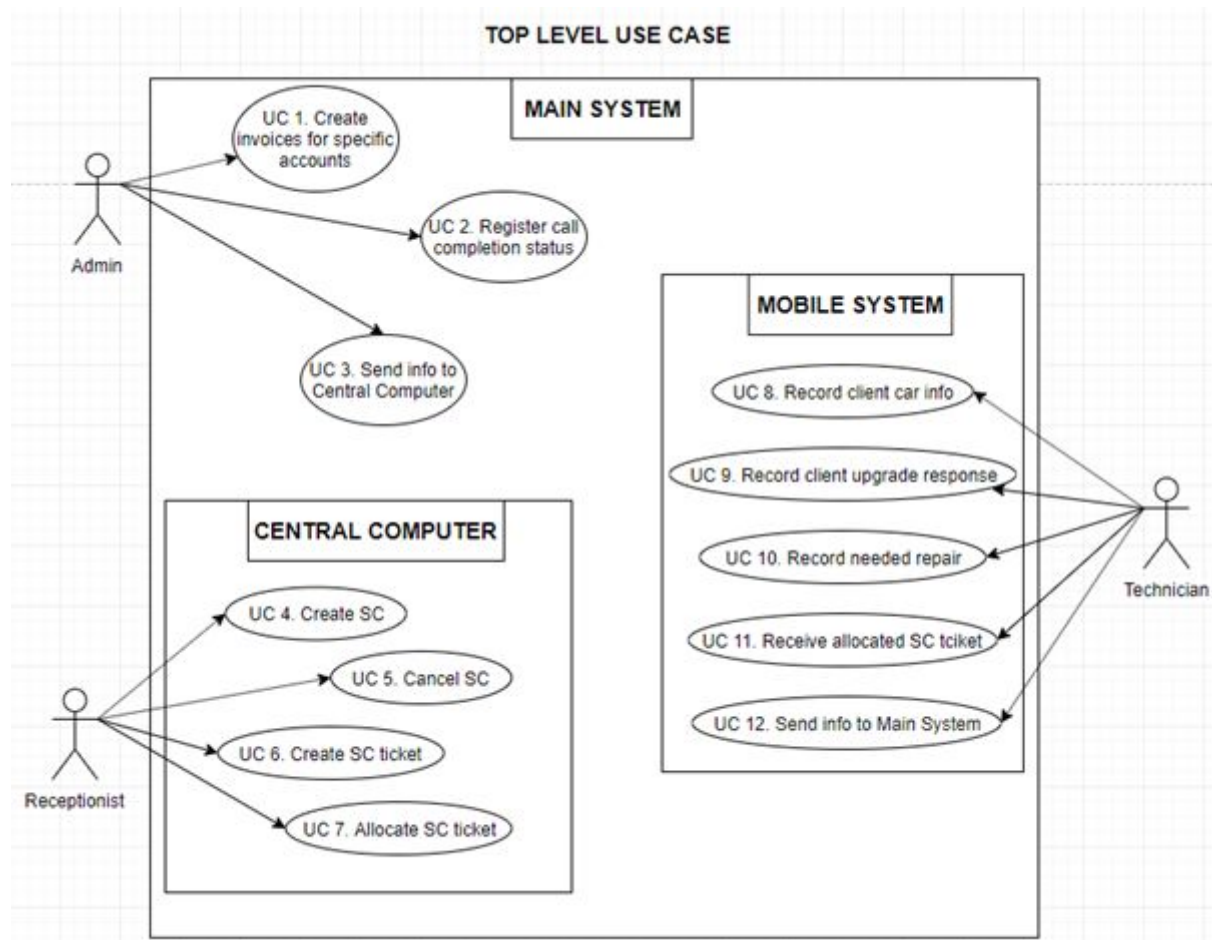
This leaves Administration (Admin) free to do their role. They are night shift workers who handle invoices and create reports. We assume when the invoice is created, cost is then added up put onto the actual invoice itself. We assume the invoices are sent to Clients each night and that the Client can bank transfer or phone Reception to pay. Admin is also responsible for programming specific mobile devices to specific Technicians, no logon is required by Technicians.

Warehouse and its staff are not our concern, along with parts stock and van inventory.

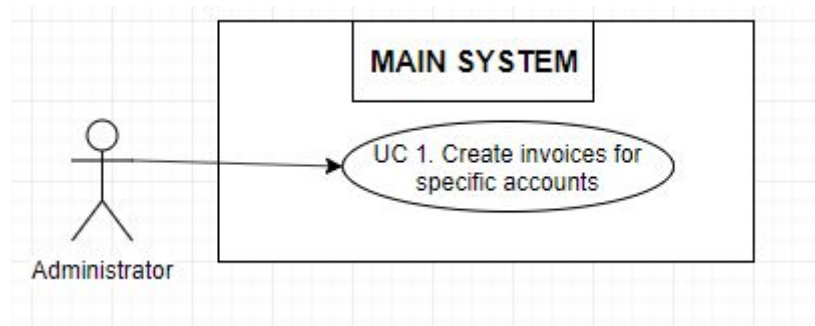
Technicians arrange their own schedule using their mobile device, it is not done automatically. Technicians have their own personal handheld device so a login is not required. Once the technician has met with the client additional information will be gathered and added to the ticket. When updating a clients account with more information it is automatically input to the system when they return to the warehouse, Admin then adds this data to the account.

B) Use case diagrams, including textual descriptions of the use cases.

Top Level Use Case

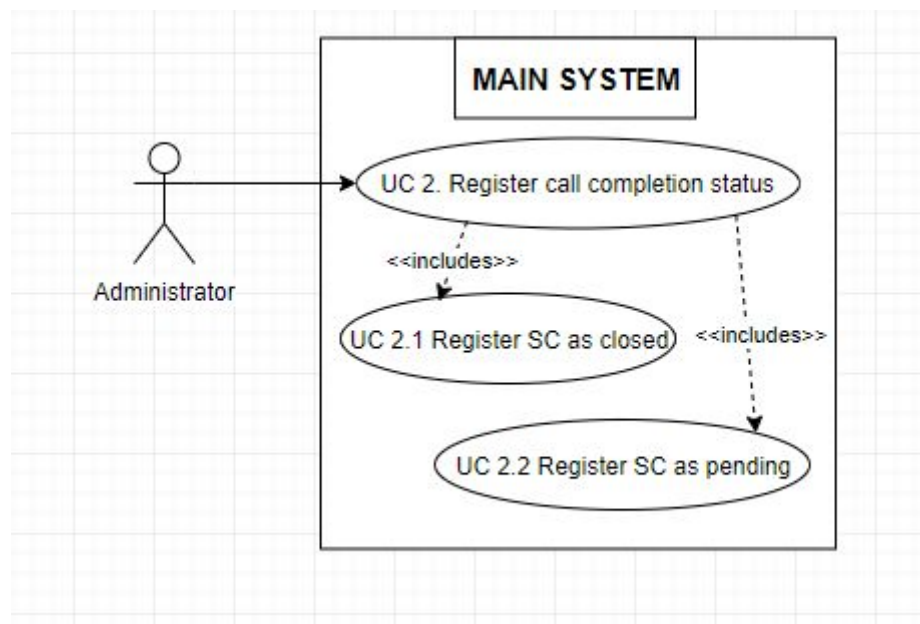


UC Diagram 1



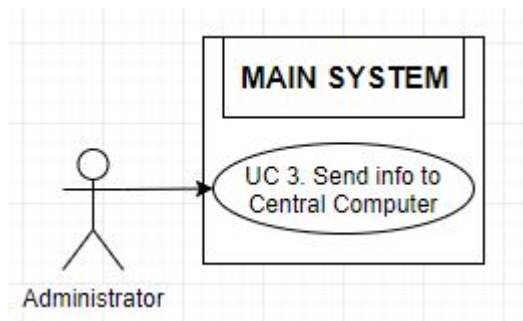
Name of Use Case	Create Invoice
Actors	Administrator
Preconditions	The account for this invoice has a completed ticket that has not been billed
Description	Admin creates an invoice linked to a specific account
Postconditions	Invoice has been created

UC Diagram 2



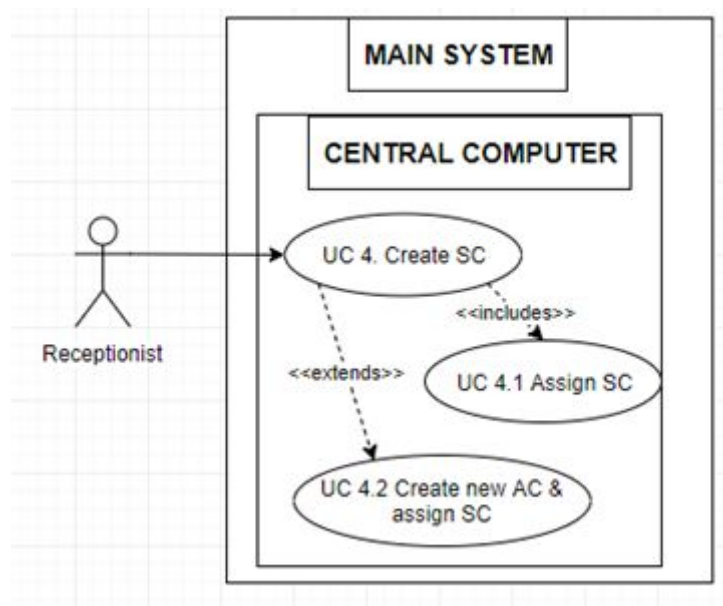
Name of Use Case	Update Call Completion
Actors	Administrator
Preconditions	A technician carried out service calls throughout the day, once they return to the warehouse all info goes to the admins
Description	Admin either registers call as closed or pending
Postconditions	Call has been closed and pending call added to the list of tickets for tomorrow

UC Diagram 3



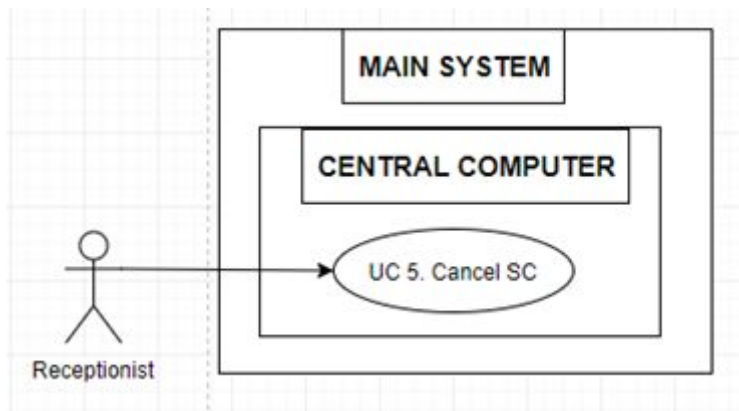
Name of Use Case	Send Data to Central
Actors	Administrator
Preconditions	Admin has unsent data
Description	Admin sends all data from main system to the central computer
Postconditions	Data has been sent

UC Diagram 4



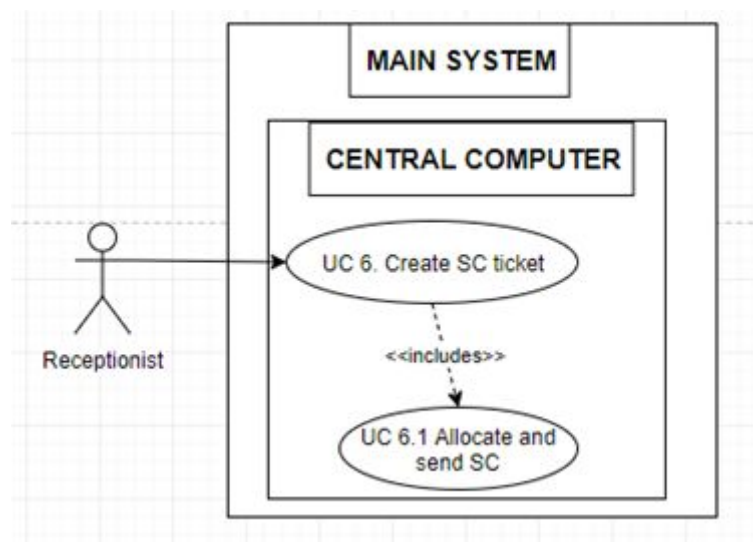
Name of Use Case	Create Service Call
Actors	Reception
Preconditions	Client phones to request service call
Description	Reception takes the clients details and creates a ticket, reception links an account to the ticket, if the client has no account then an account is created now
Postconditions	Service call is created and assigned to a client account

UC Diagram 5



Name of Use Case	Cancel Service Call
Actors	Reception
Preconditions	Client phones wanting to cancel a service call
Description	Reception cancel the service call related to that client account
Postconditions	Service call has been cancelled

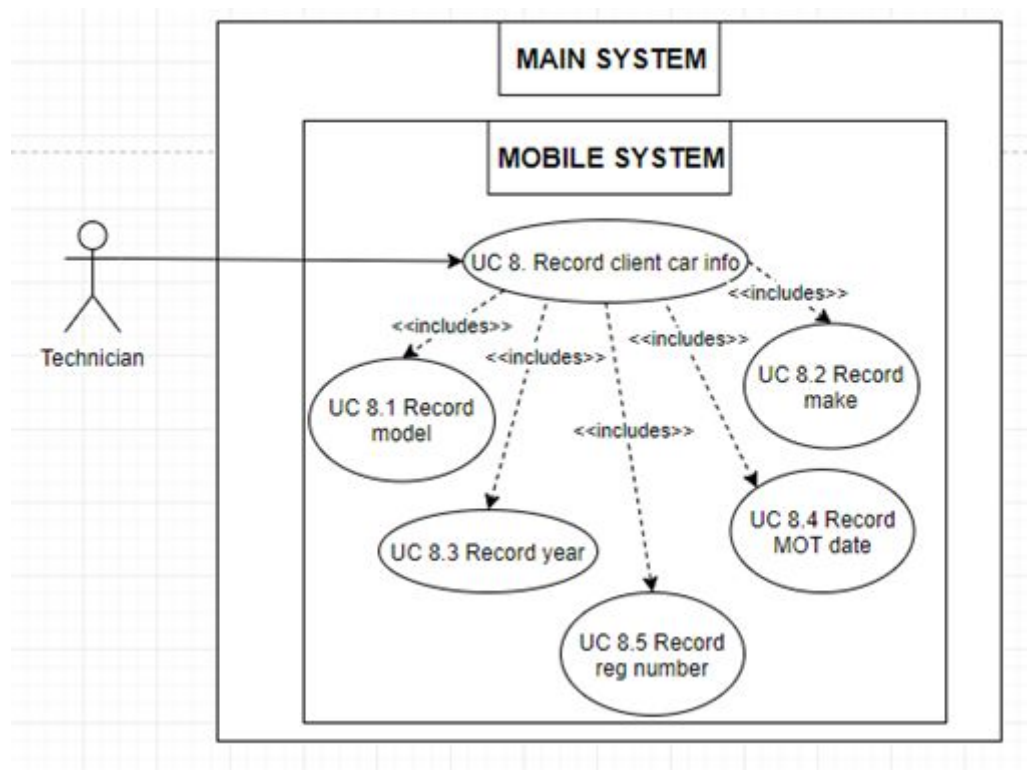
UC Diagram 6



Name of Use Case	Create Ticket
Actors	Reception
Preconditions	A client has phoned to request a service call
Description	Reception creates a ticket and it is automatically allocated to a Technician
Postconditions	Technician has a new service call ticket

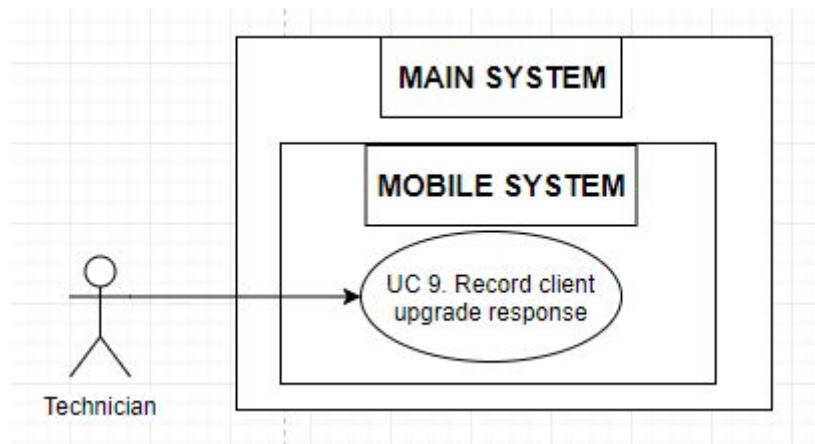
Also, we decided that UC 7 works better as 6.1. Allocation wouldn't happen without creation.

UC Diagram 8



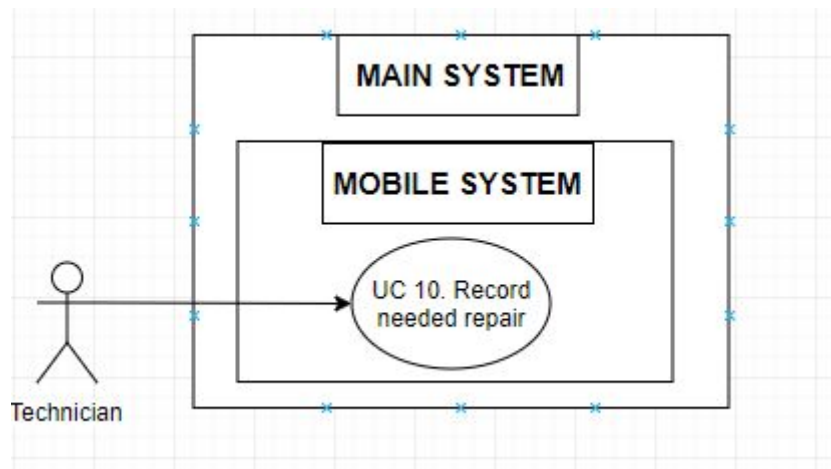
Name of Use Case	Record Car Info
Actors	Technician
Preconditions	A new client requested a service call, the Technician arrives to the client
Description	Technician records the clients car info: Model, Make, Year, Reg and MOT date
Postconditions	Client's vehicle details have been updated

UC Diagram 9



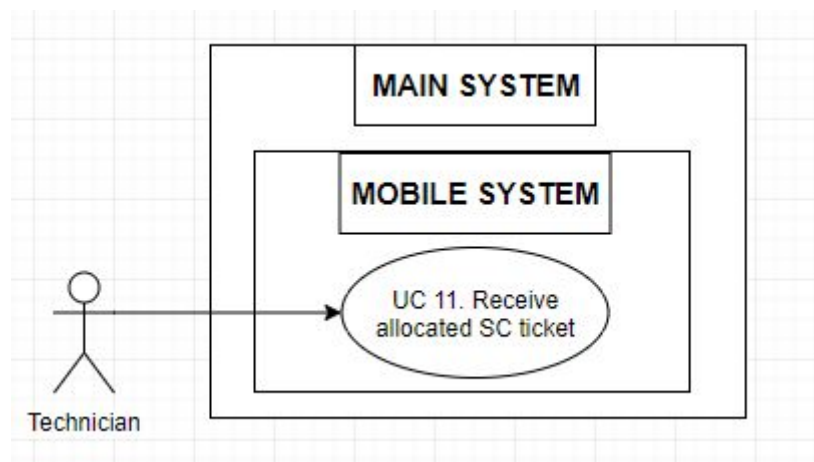
Name of Use Case	Record Upgrade Response
Actors	Technician
Preconditions	Technician has arrived to a client, the client doesn't have the best contract
Description	Technician asks the client if they would like to upgrade their contract, records the response
Postconditions	Response is recorded

UC Diagram 10



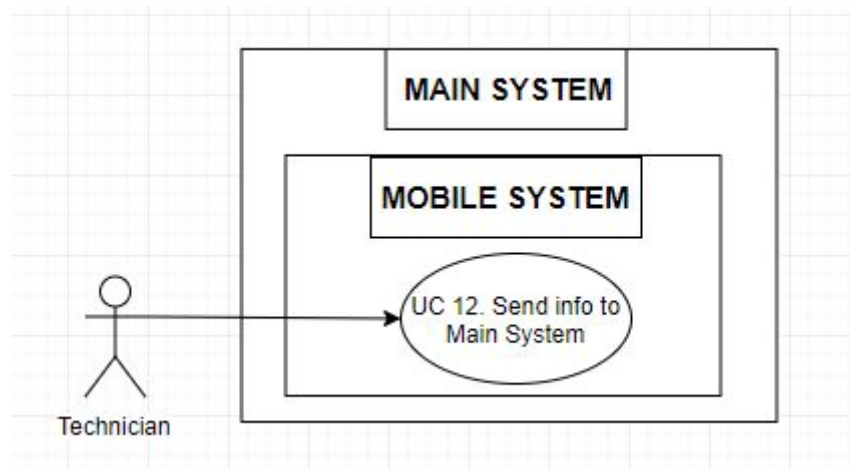
Name of Use Case	Record Repair Needed
Actors	Technician
Preconditions	Technician arrives with a client.
Description	Technician records what repairs are needed
Postconditions	Ticket is updated with what repairs are needed

UC Diagram 11



Name of Use Case	Receive Ticket
Actors	Technician
Preconditions	A client has phoned Reception and a ticket with related service call have been made, ticket has been allocated
Description	Technician receives the service call ticket on their mobile device
Postconditions	Ticket received

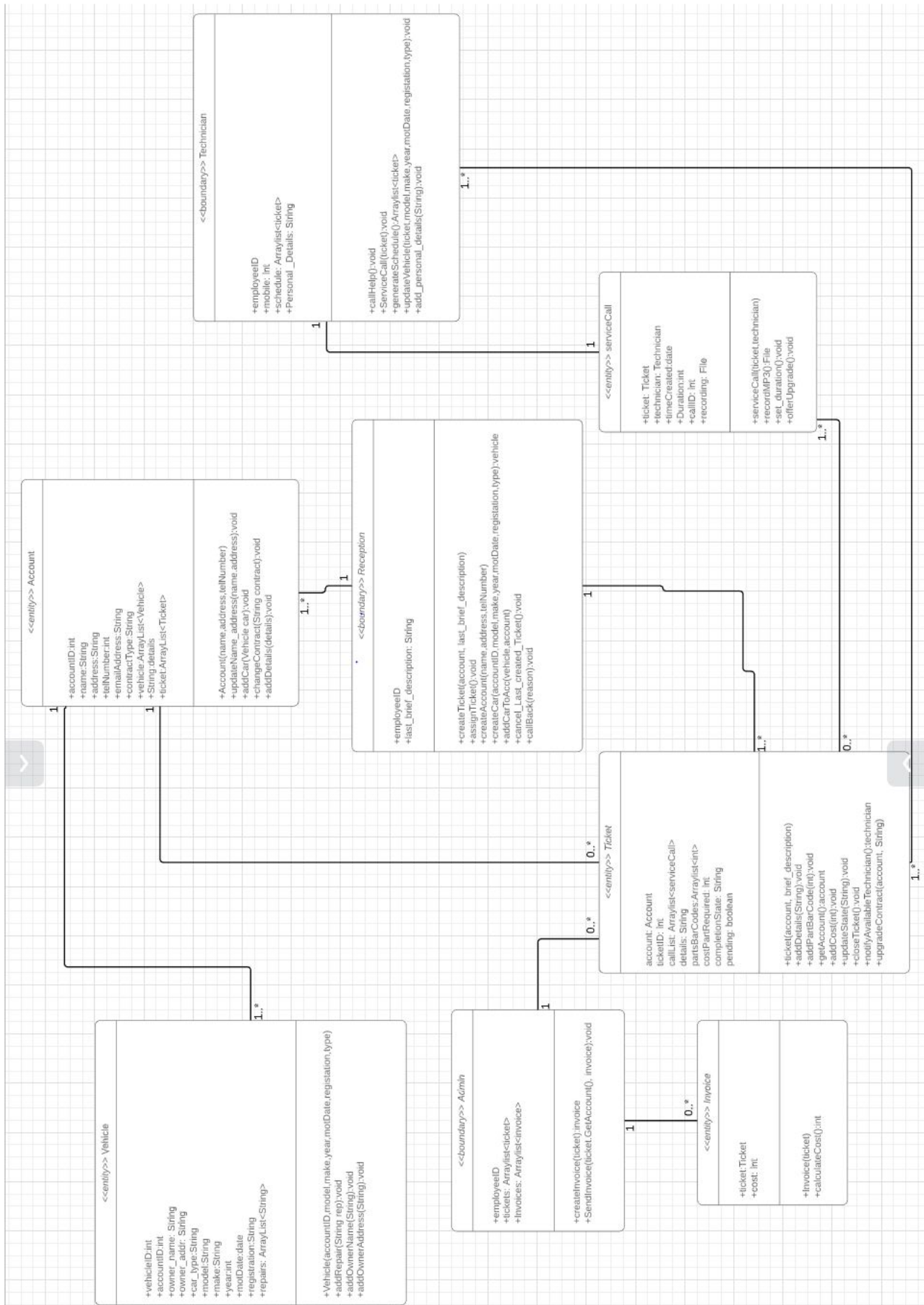
UC Diagram 12



Name of Use Case	Send Info to Main
Actors	Technician
Preconditions	Technician returns at the end of the working day
Description	All of the technicians info for the day is sent to the main system (admin)
Postconditions	Info sent

Part 2

A) Static Design Model



Description of classes:

Account

This class is the class which is needed for all clients. Accounts are needed to record significant information about clients and their vehicles. Each account has its own unique AccountID which is int, name, address, telephone number, email address, list of all vehicles an account possesses, contract a client has bought and other details. In addition to that, it carries the list of all tickets related to it inside ArrayList<ticket> variable. Each account can be modified:

- Name and address can be edited using method updateName_address (name, address)
- More vehicles can be added to 1 account using method addCar(vehicle)
- Account contract type can also be altered using changeContract(String) method
- More details can be added using addDetails(String)

When a client calls, reception either creates a new ticket and associates the client's account with it or, if client is calling regarding an unsolved ticket, they are redirected to a technician.

Reception

The reception's role is obvious - it acts as a regular reception. When a client calls, their problem is recorded into String last_brief_description variable which then is used to create the ticket. Also, since receptionist is an employee they have to have their own employee ID which is recorded into the corresponding field.

Reception is mainly responsible for:

- Creating accounts using createAccount(name,address,telNumber) method
- Creating tickets using createTicket(account, last_brief_description) method

In addition to that, they can also gather information about client's vehicle and call a particular client back if there is a reason or in case of a prank call they can cancel tickets.

Technician

Technicians do all the dirty work. Since each of them carries a mobile device each of them has to have their own unique telephone number and employeeID. Both of these are INTs. They can also have personal details recorded into their profiles (field Personal Details) using add_personal_details(String) method. Main responsibilities of a technician include:

- Generating their own unique schedule for a day using generateSchedule() which would record the new schedule into the schedule(arraylist<ticket>) field
- Updating clients car information using updateVehicle()
- Updating ticket completion progress using updateState()
- Offering new contracts to clients and if they accept it, upgrading their existing contract using upgradeContract(String)

If there is a problem a technician needs help with they can call for help using callHelp().

Invoice

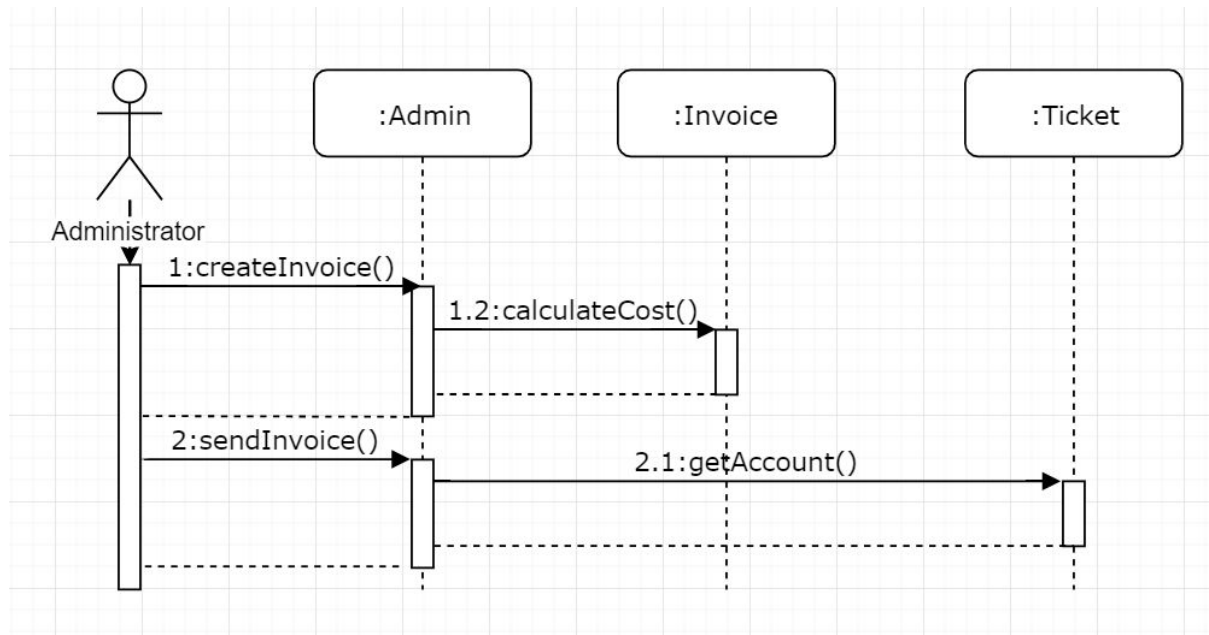
The main purpose of invoice class is to create invoices which are directly related to tickets and then calculate the cost of all services provided to a client using calculateCost() method. Invoice generation is done by the admin team, they are also responsible for sending invoices to accounts. Invoice class contains 3 fields:

- int cost
- Account acc
- Ticket ticket

Account and ticket are there to connect a particular invoice to a particular ticket and, therefore, client.

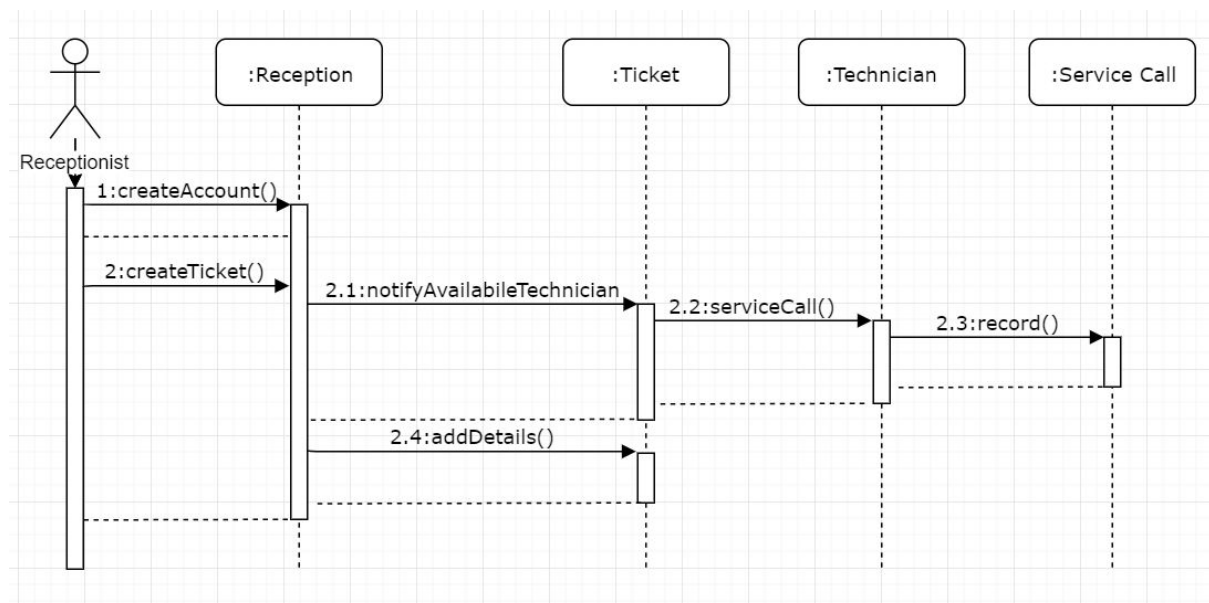
B) Dynamic Design Model - Sequence Diagrams

Create and Send Invoice



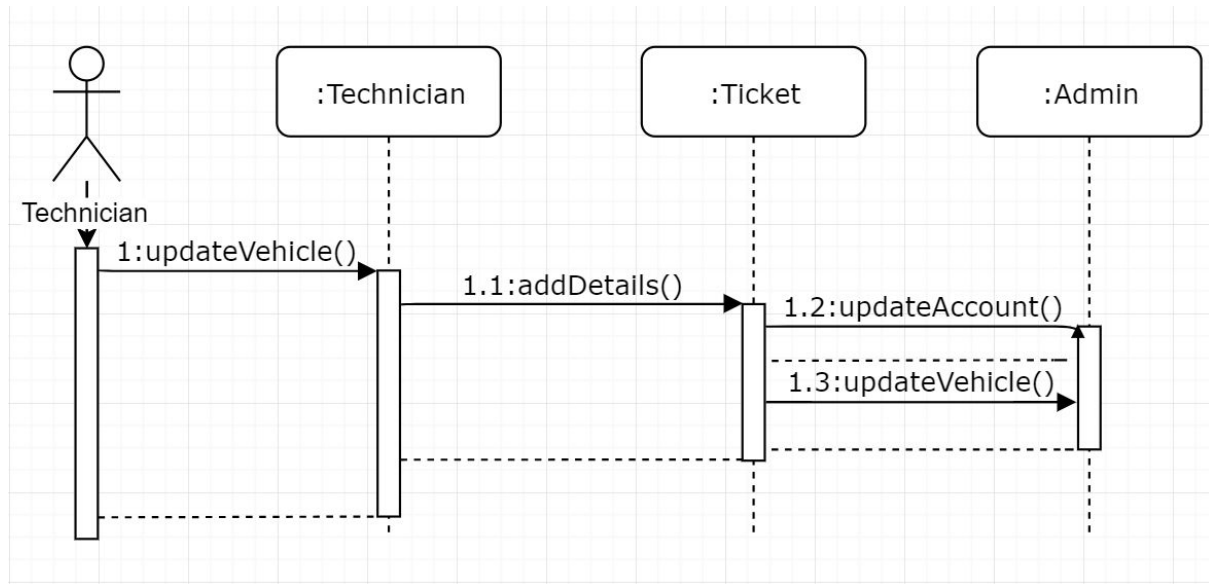
Here we see an administrator creating and sending an invoice. The invoice calculates its total cost and then sends the invoice out to person with the corresponding account which can be found in the ticket.

Create Ticket/Service Call



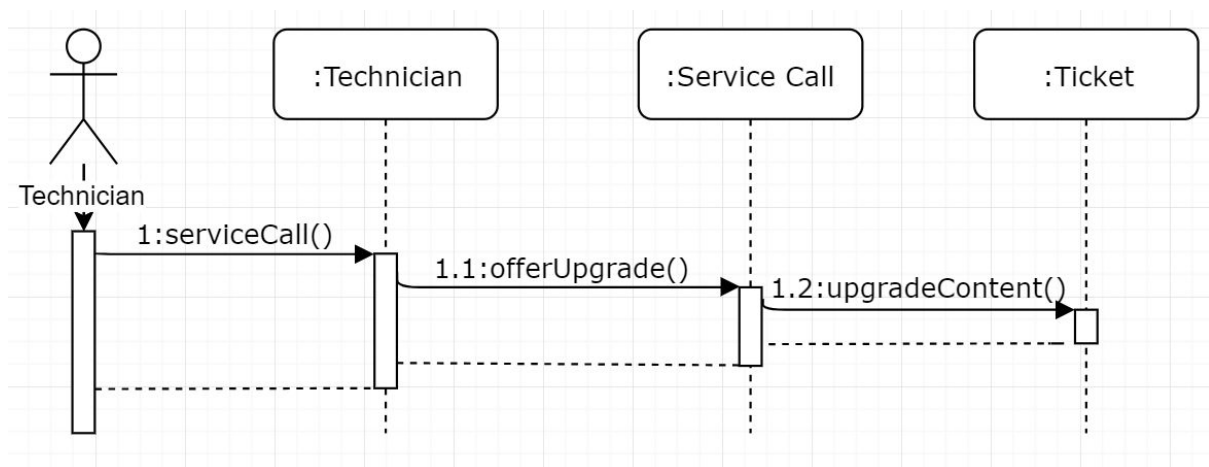
Here the receptionist is creating a ticket, this will create an account for the user if one does not already exist. The initial ticket is created and then the technician is notified through their mobile system. They then conduct the service call where the call is recorded for later use. Afterwards any additional information is updated to the ticket.

Technician Records Clients Car Information



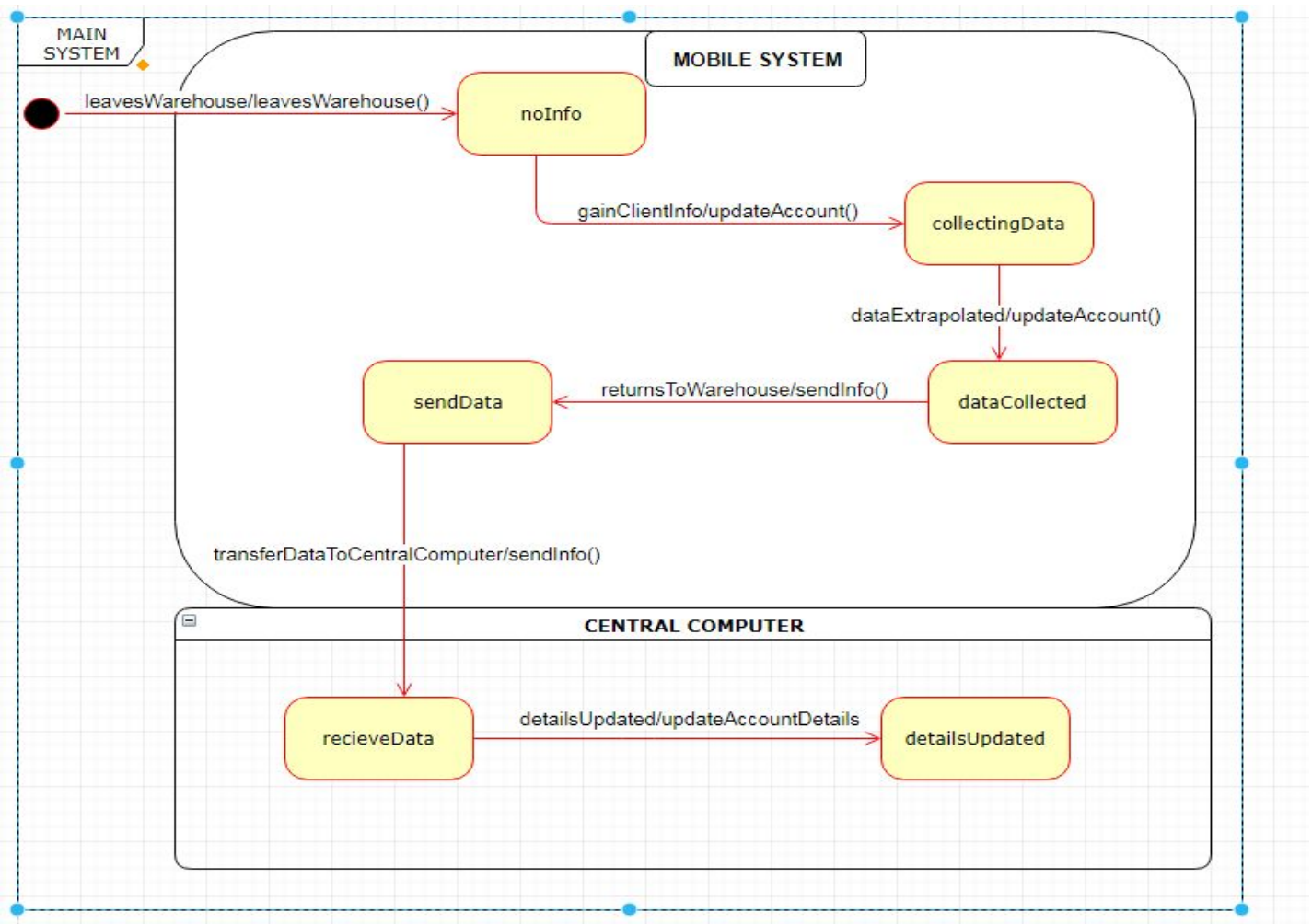
Here the technician is recording the second hand information gathered from the client. The technician will add the new details to the ticket and then the administrator takes this updated ticket as they require it to generate an invoice, and update the account.

Technician Records Clients Upgrade Response



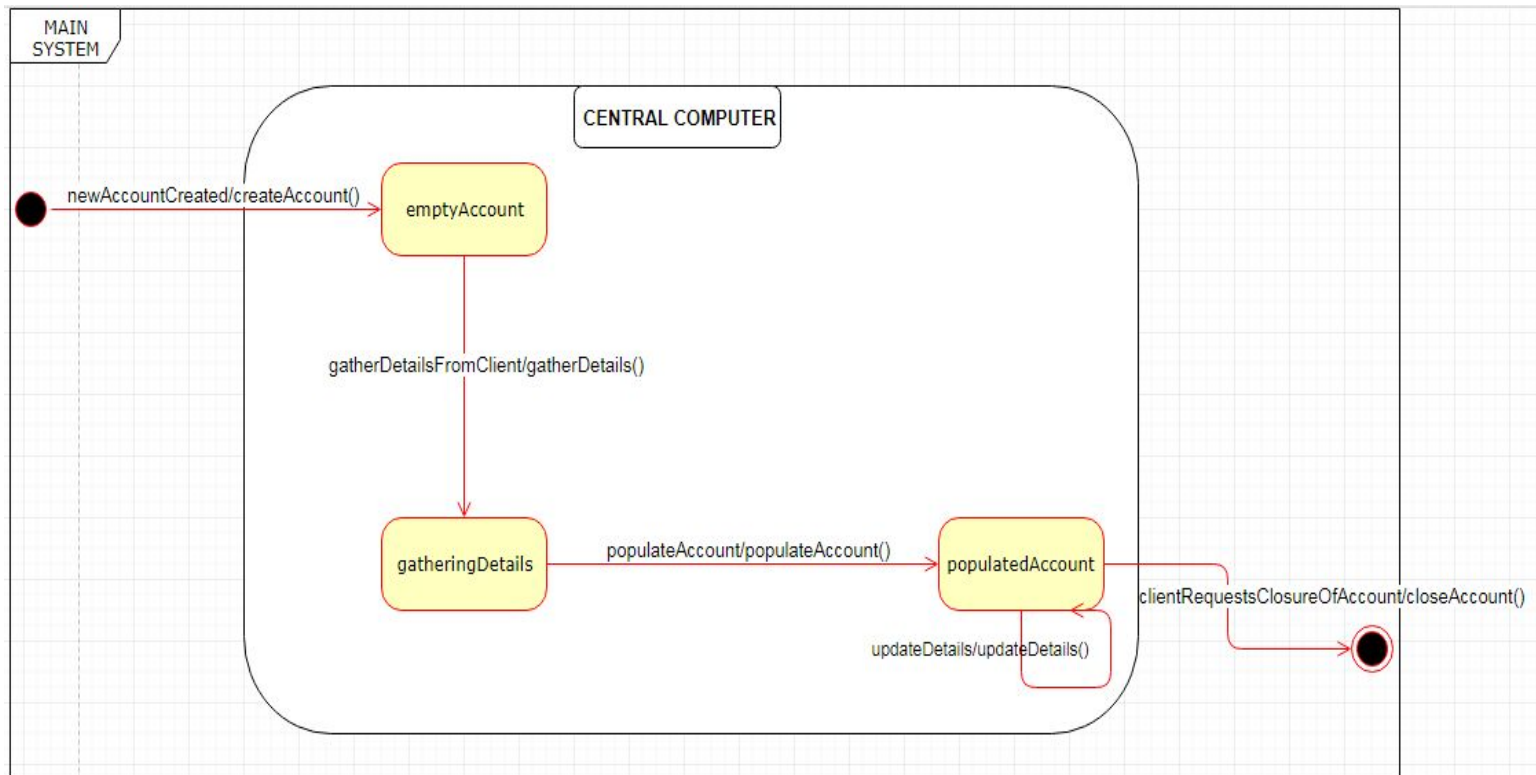
Here the technician is offering the client to upgrade their contract with SafeJourney. During the service call the technician makes the offer and the relevant response updates the content of the ticket.

Dynamic Design Model - State diagrams



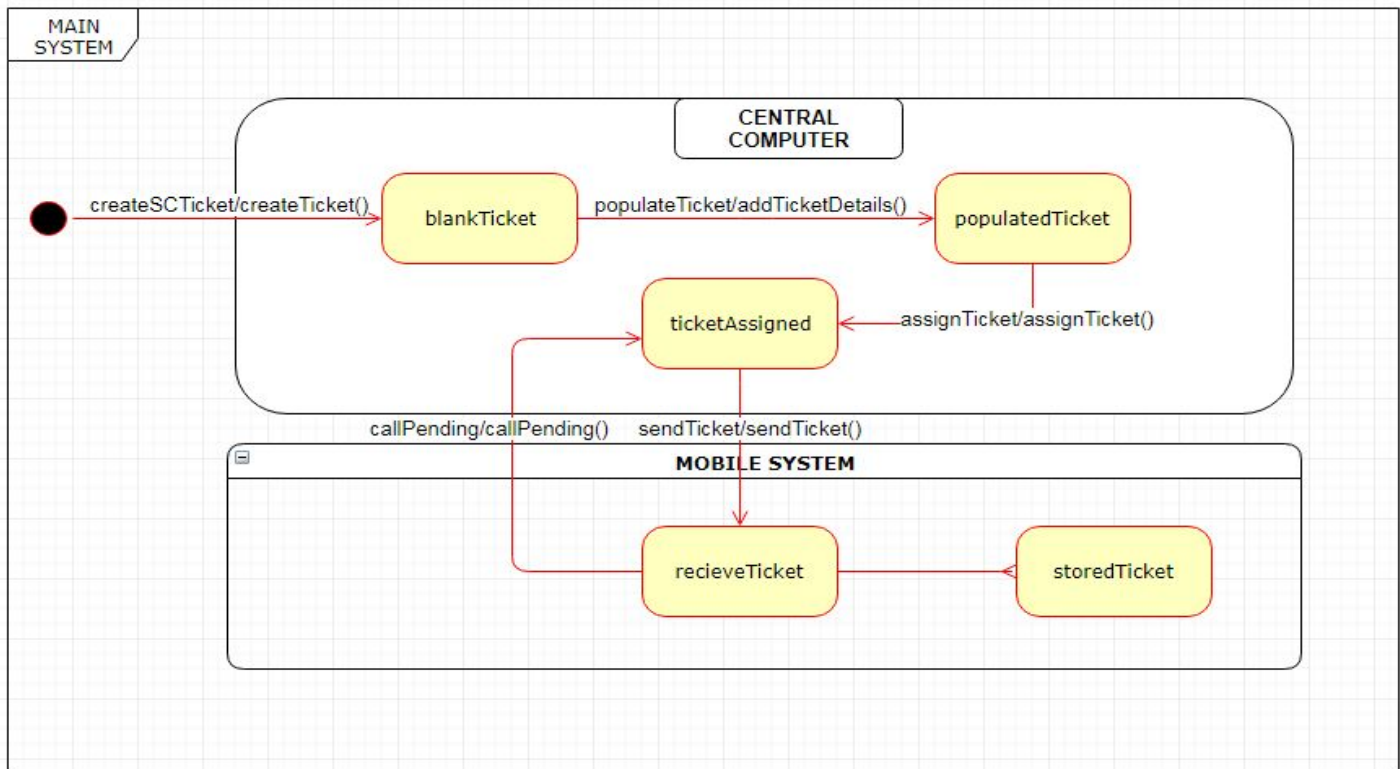
Updating new customers account details

In the main system, the technician start to leave Warehouse without any clients information. When we have new clients, we will collect all the relevant data and create a new account for updating their information. If the client is not new then the account corresponding to that client is updated. After the technician has the information it is sent via the mobile system into the central computer system where the details are updated.



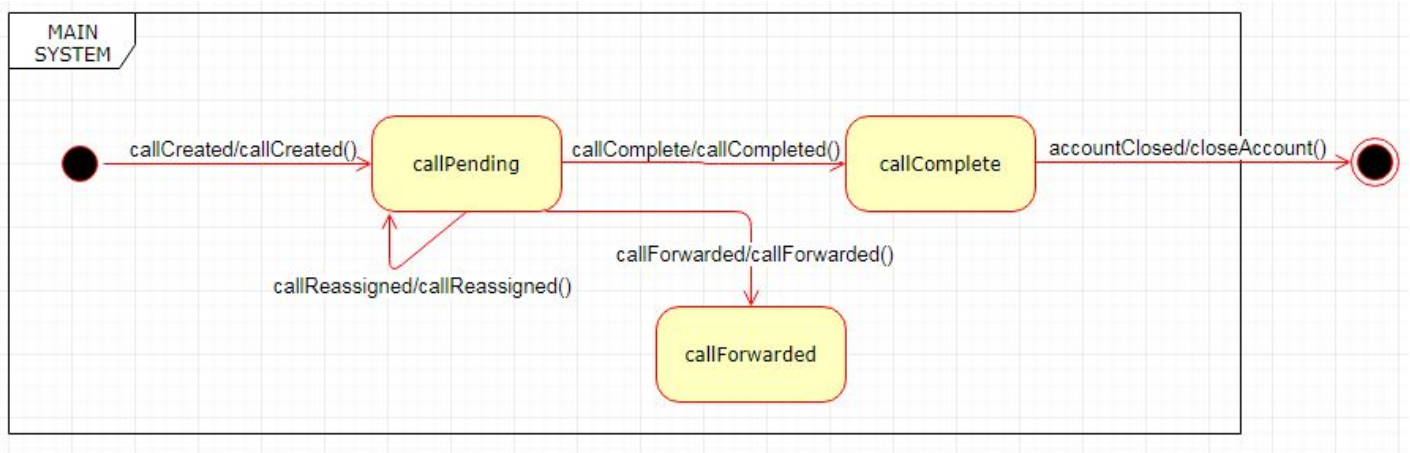
Create new account

When the receptionist creates a new account for a client they must first create a blank template of an account, afterwards the receptionist gathers the data collected and populates the empty account. From here the populated account can be updated with data that the administrator or technician has, or the account can be closed if the client phones reception to do so.



Ticket completion state

Here the receptionist creates a service call ticket and then populates it with information gathered from the client such as a brief description of the problem. This ticket is then assigned to a technician from the central computer system to the mobile system. From here the ticket can be stored for use throughout its lifetime being connected to the job, or the ticket can be issued back to the central computer system as pending. The administrator reviews all ticket completion states at the end of the day and pending tickets are to be reassigned next day to technicians for completion.



Service call state

A service call is automatically created when a ticket is created by reception. The service call is initially pending and will remain pending until updated by the technician as forwarded to their sister company or completed by that technician. Service call data will be stored in the system until the account it is linked to is requested to be closed by the client, then all related data to that account is deleted.