

Contents

1. Executive Summary	3
2. Requirement specifications	4
I. Requirements	4
II. Assumptions	6
III. Use Cases	7
3. System Design	
I. Static Design	12
i. Class Diagram	12
ii. Textual Description	13
iii. Class stereotypes	14
II. Dynamic Design Model	
i. Sequence Diagrams	15
ii. State Diagrams	20

1. Executive Summary

The given problem stated that CompuFix has a need to update their system to accommodate a rise in devices needing repair due to an explosive growth in home and mobile computer systems. The system we propose is designed to work seamlessly in conjunction with CompuFix's currently daily working practices and the existing client database. The major change from the previous system is in how recording service requests are recorded, how jobs are assigned, how necessary job information is gathered and how invoices are processed. Previously client information was manually recorded by Customer Relations Staff to be later updated by administration overnight; our proposed system will offer operations to enable data the system to automatically process the data from each working day.

The system we propose is comprised of two parts: the main system and the mobile devices. The main system handles the creation and updating of a client account, the creation of an invoice after a job is completed, including all cost calculations, to the eventual creation of a quality control report. The mobile device system will enable the CompuFix mobile staff i.e. engineers and warehouse staff to interact with the system remotely. It allows an individual engineer to log in and out of a single mobile device which will provide their allocated jobs list along with the facility to update the job records and schedule any appointments if desired. In addition, the engineers may use their assigned mobile devices to forward off-site jobs to the sister company, MajorFix. This system is designed to provide a long term solution which will maintain functionality if the workload were to be increased by a further rise in the services requested from CompuFix.

2. Requirements Specification

I. Requirements

Functional Requirements

1. Database Access API – DAAPI
 - a. Access the original client database
 - i. System must be able to create a client account which includes:
 1. Name recorded
 2. Address recorded
 3. Telephone number recorded
 4. Unique ID number created
 - ii. Record service history
 - iii. Search the database for an existing client account
 - iv. Store a record of a client's service request
2. Automated Overnight Administration Process
 - a. Identify barcodes from an existing database
 - b. Allocate and reallocate jobs
 - c. Update details for an individual job record
 - d. Issue invoices
 - i. Calculate the cost of completed jobs
 - e. Update mobile devices with allocated job lists
 - f. Update existing client accounts
 - g. Update maintenance contract level
 - h. System must be able to issue new job records for:
 - i. Incomplete jobs
 - ii. Partially completed jobs
 - i. Create a quality control report
 - j. Send a confirmation email to the clients
 - k. Produce a spare part lists for incomplete and partially complete jobs
3. Mobile Devices
 - a. Engineer must be able to enter client details collected during the visit
 - i. Client email (if applicable)
 - ii. Summary of computer equipment
 - b. Engineer must be able to input visit summary
 - i. Visit duration
 - ii. Spare parts used
 - iii. Mileage

- c. Engineer must be able to edit job status
- d. Push job data back to main system
- e. Send off-site jobs to MajorFix

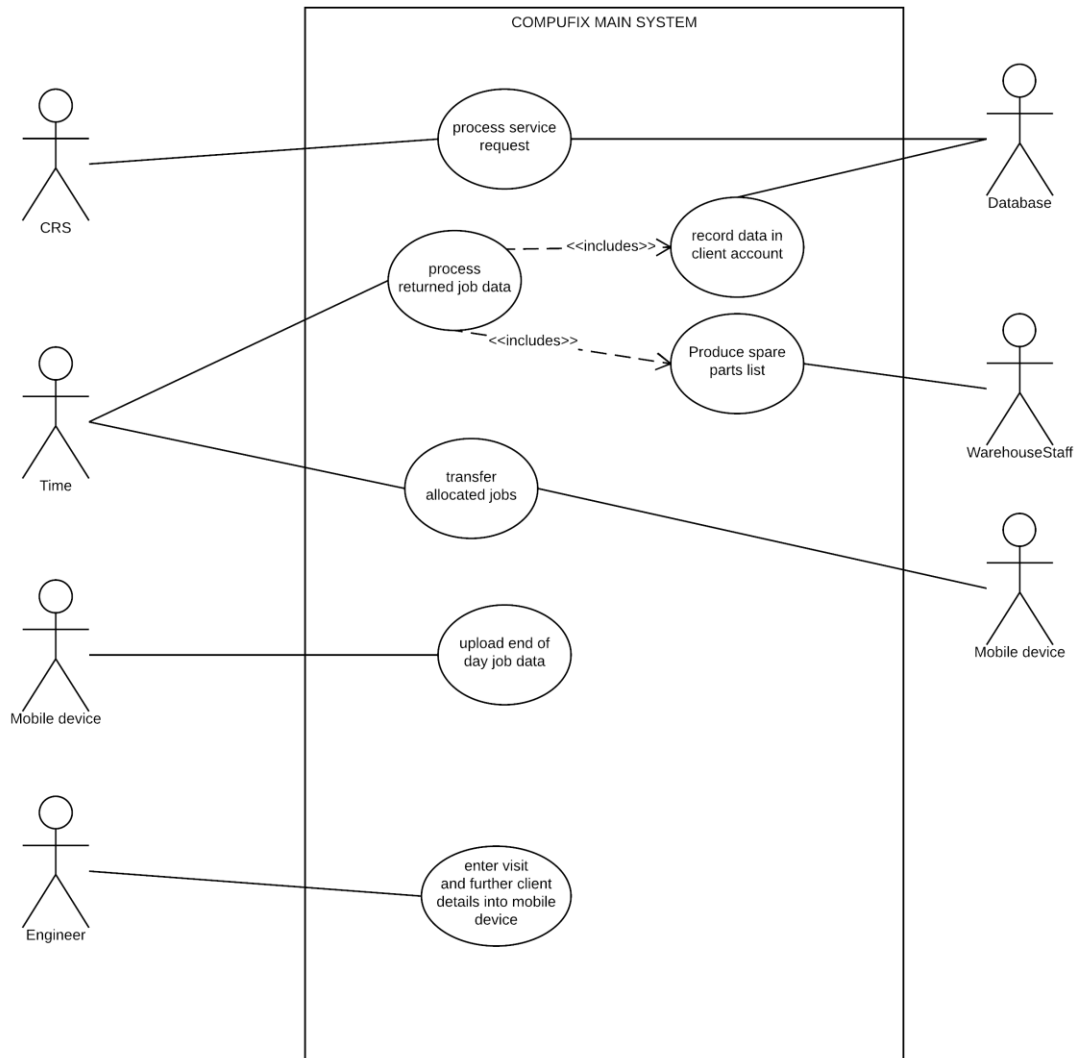
Non-functional Requirements

1. Main System
 - a. Only complete job records are added to a clients' service history
2. Mobile System
 - a. Prompt engineer to offer maintenance contract upgrade on completion of summary
 - b. A random mobile device is assigned to each engineer and warehouse staff member every morning

II. Assumptions

- i. There is one main system but many Customer Relations Staff(CRS) therefore it is possible to have multiple CRS interact with the main system through individual UIs.
- ii. The customer's maintenance level has been stored as an integer. If the customer has no contract level it is stored as a 0, a level 1 contract is stored as a 1 and a level 2 contract is stored as a 2.
- iii. The processing data manager only cares about starting/stopping the rest of the operations depending on the time so if there is a problem with an operation not starting at the right time then the processing manager would most likely be responsible
- iv. We assume that our system will keep a system log in case something goes wrong.
- v. Each staff member has a unique payroll number which can be used to uniquely identify them within the system.
- vi. Barcodes are provided and the mobile devices identify and store them in the database
- vii. Mobile devices are generic but members of staff can log in to them
- viii. Each member of staff that is given a mobile device has a staff ID number that can be used as a PIN to log in to the device.
- ix. There can only be one active job per client – the rest of the closed (completed or otherwise) job records are stored in the service history associated with each client's account.
- x. Parts are only scanned by engineers when they are going to be used.
- xi. There is a list of barcodes stored in the system which is loaded into the mobile device memory
- xii. The confirmation email sent to the client contains the job details

III. Use cases



Use case: Process service request

Brief description: CRS receives a service request from a client and processes it using the system

Actors: CRS, Database

Preconditions: Client must initiate a service request with Customer Relations Staff via telephone with CompuFix

Normal Flow:

1. The CRS initiates the process of a new service request.
2. The CRS retrieves an existing client from the database.
3. The CRS proceeds to create job record

Abnormal Flow:

4. The client requesting service does not have an existing account
5. A new account is created in the database

Use case: Transfer allocated jobs

Brief description: In the morning before the Engineer leaves the base, the allocated jobs are transferred to the mobile devices

Actors: Time, Mobile Device

Preconditions: Jobs have been allocated to the Engineers

Normal Flow:

1. Main System pushes client job records to the mobile devices at the end of the previous working day
2. Each mobile device receives its allocated job list

Abnormal Flow:

1. Main System has no client job record to push to the mobile devices

Use case: Upload end of day job data

Brief description: Data collected by the Engineers during the working day is uploaded to the Main System

Actors: Mobile Device, Location

Preconditions: Data has been collected by the Engineer during the working day

Normal Flow:

1. Data collected by the Engineers is pushed to the Main System from the mobile devices when the Engineer returns to the warehouse

Abnormal Flow:

1. No data has been collected by the Engineer

Use case: Process returned job data

Brief description: Data returned by the Engineers is processed by the Main System

Actors: Time, Database, Warehouse Staff

Preconditions: Data returned by the Engineers has been successfully received by the Main System

Normal Flow:

1. Status of job is checked – job is complete
2. Cost of the job is calculated
3. Invoice is generated
4. Confirmation email is sent to the client
5. Quality Control report is generated
6. Job is added to the client's service history

Abnormal Flow:

1. In step 2 of the normal flow, the job status is incomplete
2. Spare parts list is produced
3. Job is allocated to an engineer

Use case: Record data in client account

Brief description: Client details collected during an engineer's visit are recorded into the Database

Actors: Database

Preconditions: Data has been transferred from the mobile devices to the main system when Engineers has returned to the depo

Normal Flow:

1. Extra client details collected during the engineer's visit are recorded into client's account on the main database.

Abnormal Flow:

1. No extra information to be uploaded

Use case: Enter visit and client details into mobile device

Brief description: Engineer enters the visit details and any new client details into the mobile device provided to them

Actors: Engineer, Mobile Device

Preconditions: Engineer is on-site to perform the service request

Normal Flow:

1. Engineer logs into mobile device
2. Engineer accesses the record corresponding to the current visit
3. The ExtraJobDetails are recorded by the Engineer
 - a. Mileage is recorded
 - b. Duration of the visit is recorded
 - c. UsedSpareParts are recorded
 - d. Email address of client is recorded (if provided)
 - e. Status of job is recorded
4. Engineer logs out of mobile device

Abnormal Flow:

1. The Engineer identifies the job as an off-site repair
2. The job record is immediately forwarded to MajorFix
3. Engineer updates the job status to completed

Use case: Produce spare parts list

Brief description: A spare parts list is produced for incomplete and partially completed jobs

Actors: WarehouseStaff

Preconditions: Job data has been returned and the job status has been checked

Normal Flow:

1. Spare parts list is produced for incomplete or partially complete jobs

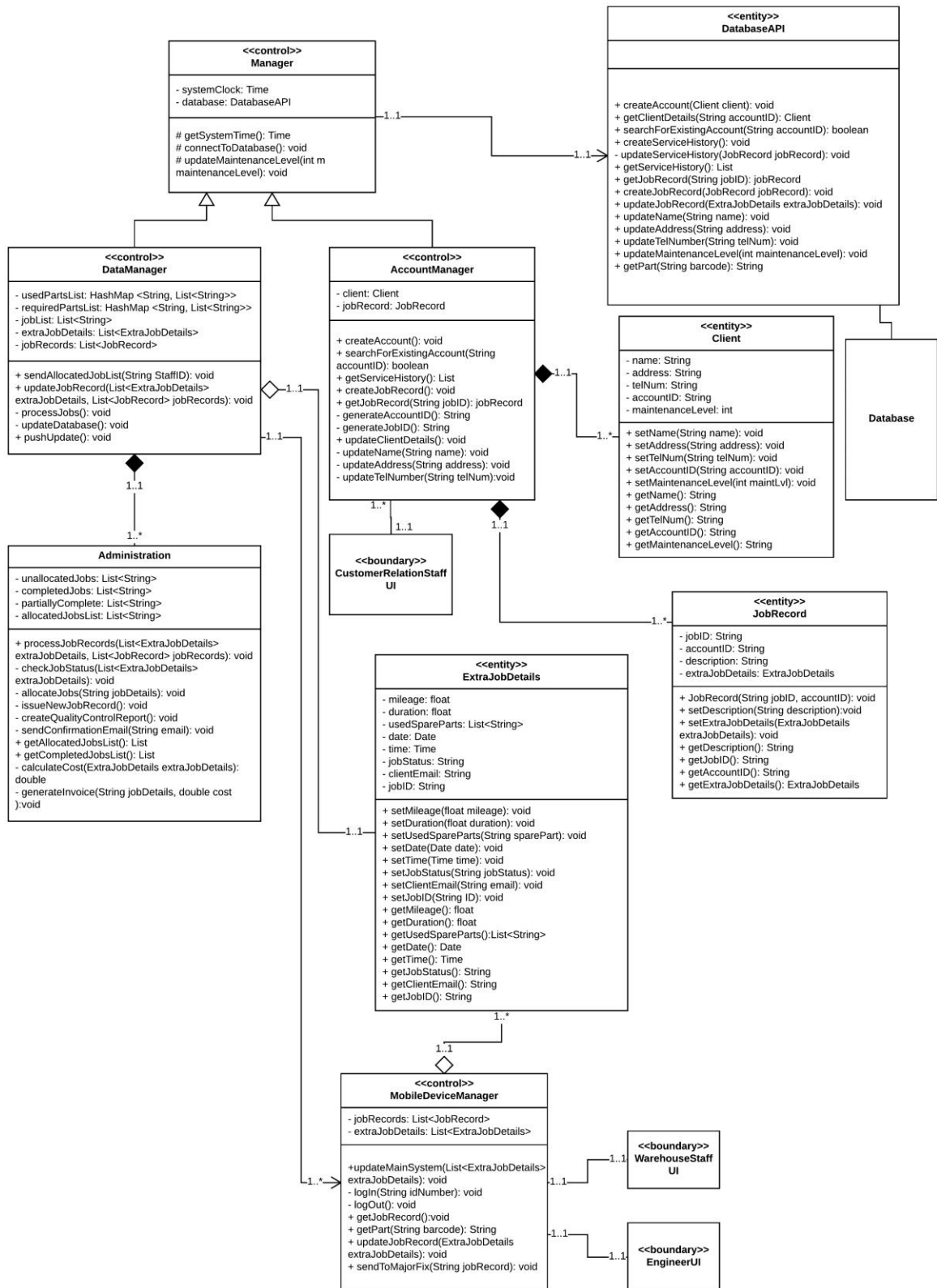
Abnormal Flow:

1. No spare parts lists are produced due to no incomplete jobs

System Design description

I. Static model

i. Class diagram



ii. Class Diagram Textual Description: CRC Cards

<u>AccountManager</u>	
Responsibilities	Collaborators
<ul style="list-style-type: none"> • Create new client accounts • Search the database for an existing client • Get a client's service history • Create a new job record • Return a specific job record • Generate new client's account ID • Update the client's details – name, address and telephone number 	<ul style="list-style-type: none"> • Manager • JobRecord • Client • CustomerRelationStaffUI

<u>DataManager</u>	
Responsibilities	Collaborators
<ul style="list-style-type: none"> • Send the allocated jobs list to the mobile devices • Update client's job record in the database • Send the updated job data to the database 	<ul style="list-style-type: none"> • Manager • Administration • MobileDeviceManager • ExtraJobDetails

<u>Administration</u>	
Responsibilities	Collaborators
<ul style="list-style-type: none"> • Process job records • Check the status of jobs • Allocate jobs to the engineers • Issue new job records for not completed jobs/partially completed jobs • Create quality control reports • Send confirmation emails to clients • Return a list of allocated jobs • Return a list of completed jobs • Calculate the cost of a completed job • generate an invoice for a completed job 	<ul style="list-style-type: none"> • dataManager

<u>MobileDeviceManager</u>	
Responsibilities	Collaborators
<ul style="list-style-type: none"> • Send updated job records to the Main System • Allow the engineer to log in/out of a mobile device • Allow the Warehouse Staff to log in/out of a mobile device • Return a client's job record • <i>Identify parts by their barcodes</i> • Update the client's job record • Send jobs which must be completed off-site to MajorFix 	<ul style="list-style-type: none"> • ExtraJobDetails • DataManager • WarehouseStaffUI • EngineerUI

iii. Short listing of boundary, control and entity classes

Boundary Classes:

- WarehouseStaffUI
- EngineerUI
- CustomerRelationStaffUI

Control:

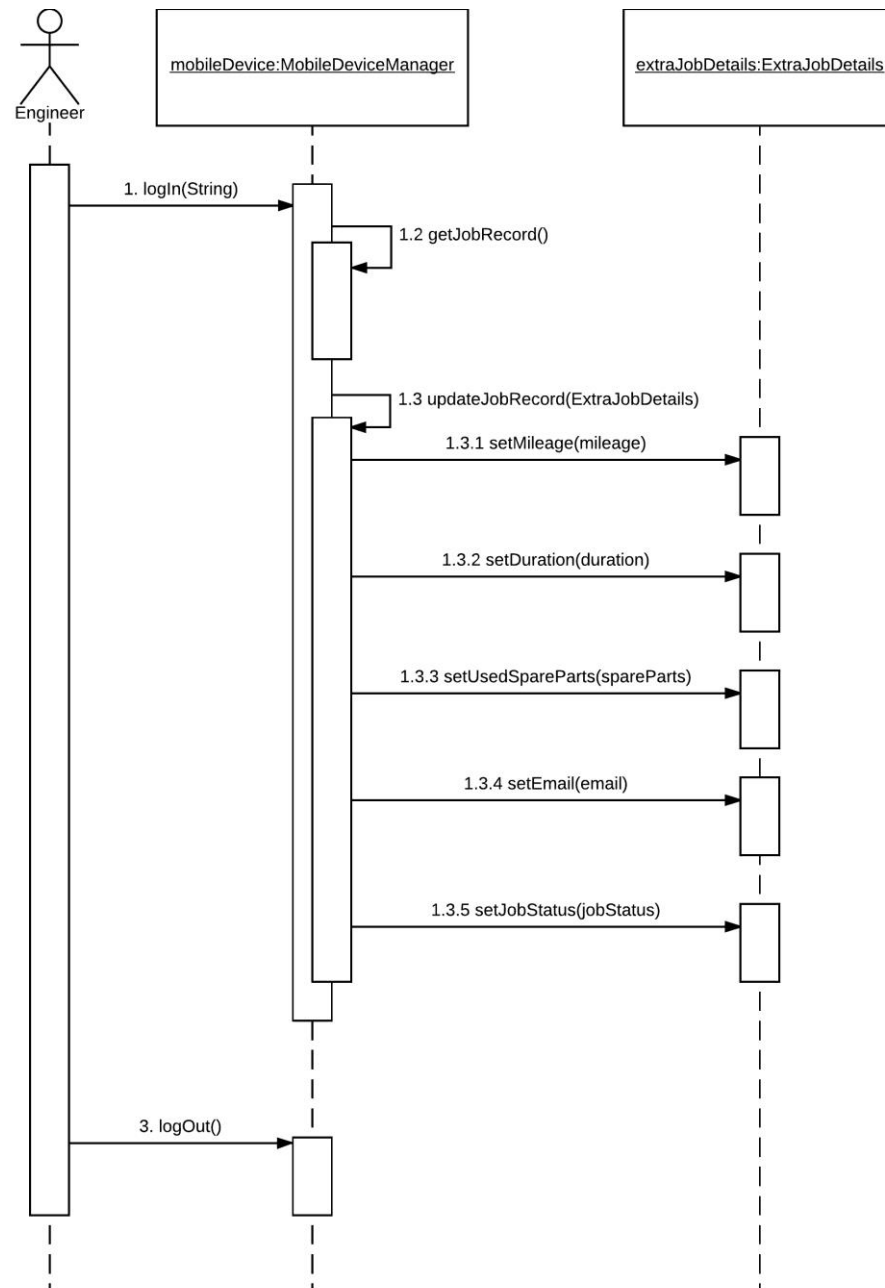
- Manager
- DataManager
- AccountManager
- MobileDeviceManager

Entity:

- DatabaseAPI
- Client
- JobRecord
- ExtraJobDetails

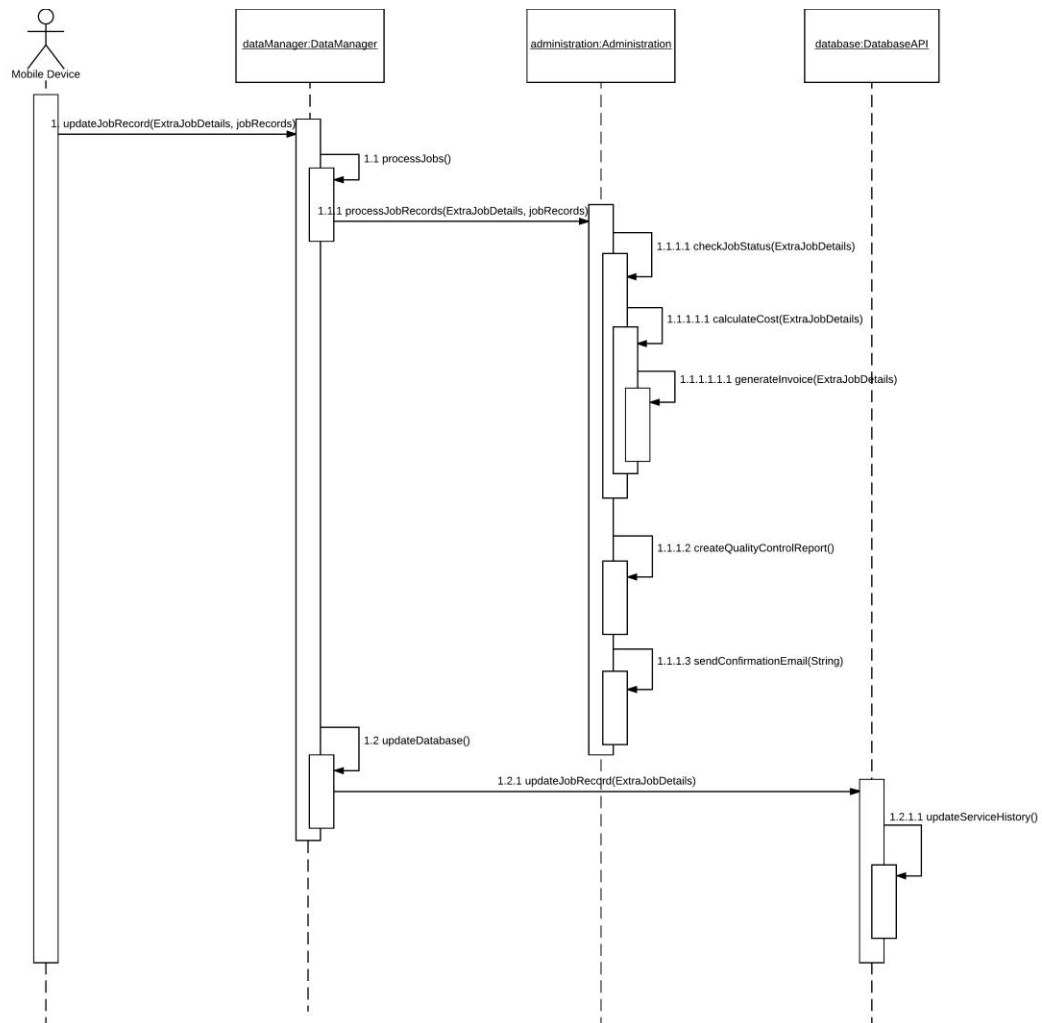
II. Dynamic model

i. Sequence diagrams



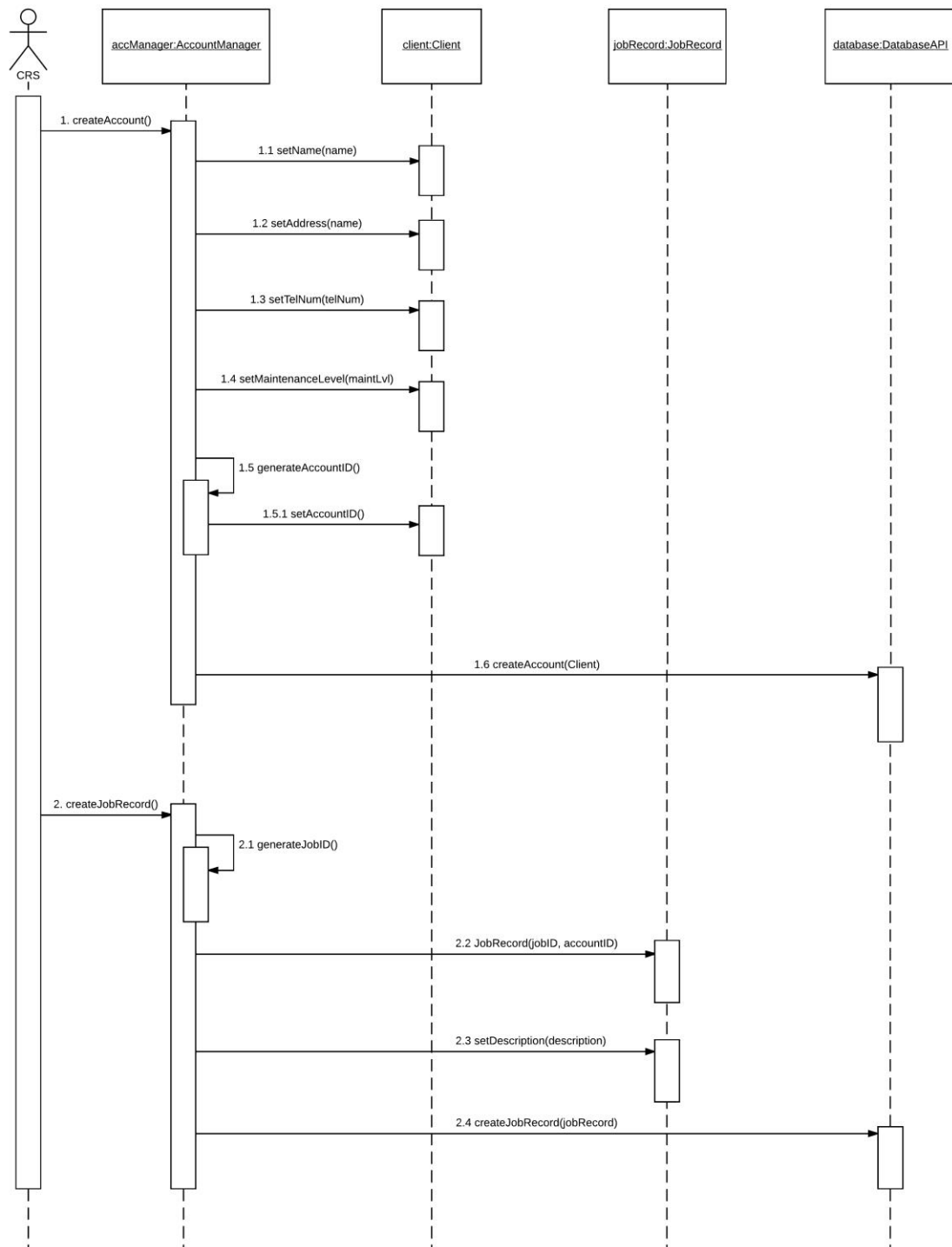
Adding Extra Job Details

This sequence diagram shows the process of an Engineer adding additional details about a job during their visit. It begins with the engineer logging into the mobile device and retrieving the record of the current job. The Engineer updates the client's job record – this information includes the mileage, visit duration, spare parts used, the e-mail of the client (if provided), and then sets the status of the job. After the Engineer has completed this, they can log out of their mobile device.



Processing Data

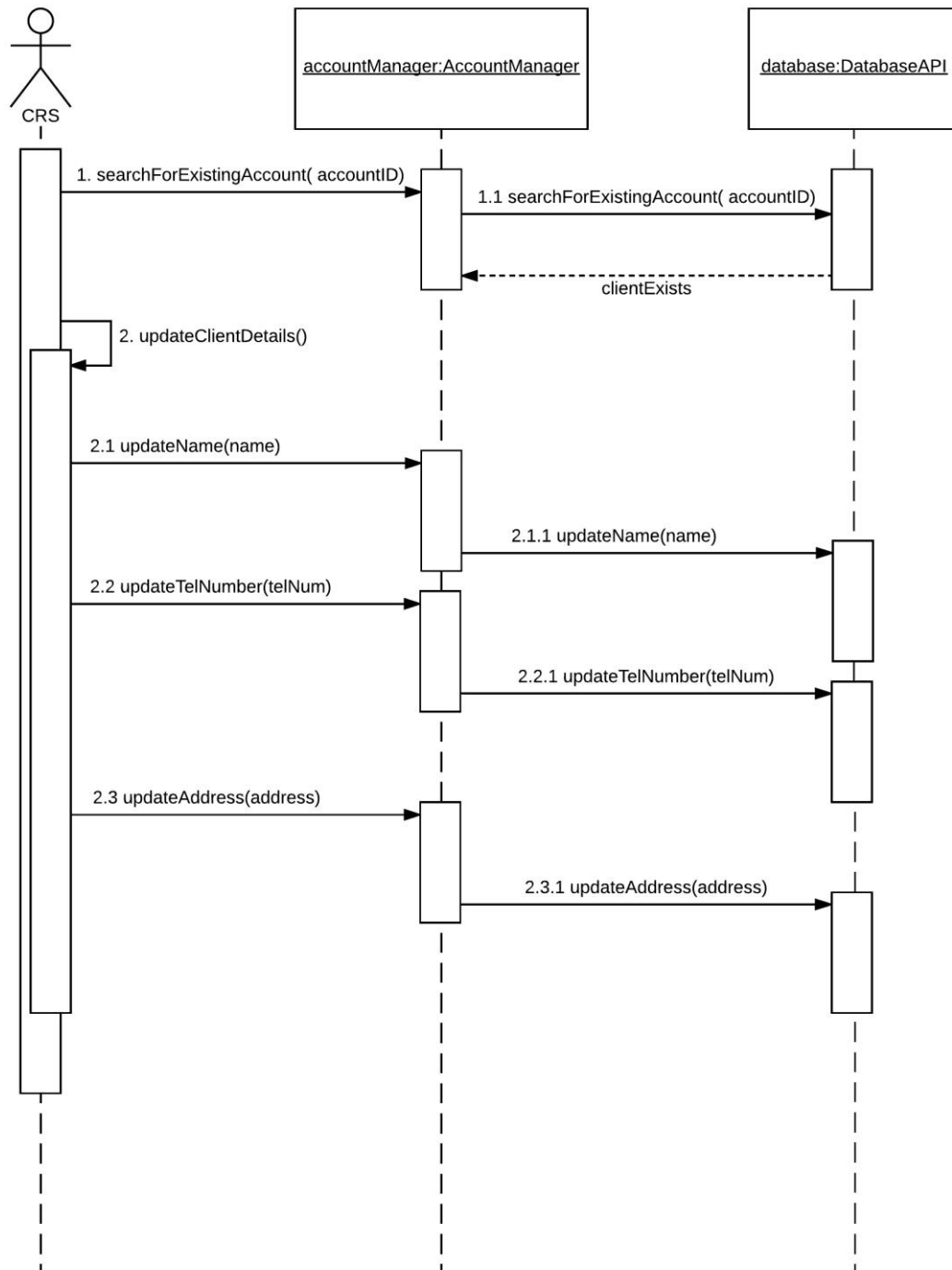
This sequence diagram shows how a completed job is processed through the night. The MobileDevice uploads the data to the Main System, specifically the DataManager class. The DataManager class creates an instance of the Administration class, and uses its operations. It calculates the cost, generates an invoice, creates a quality control report and sends a confirmation email to the client. The Main System proceeds to upload this data to the database, and updates the service history of the client.



Service Request

This sequence diagram shows the process of account creation for a new client, as well as the creation of a job record. These are both executed by the CRS. The process begins with the CRS creating a new account via the AccountManager class which in turn creates a Client class. The Client class holds the name, address, and telephone number of the client, along with the maintenance contract level on their account and the client's account ID. The client's account ID is generated by the AccountManager class. It then proceeds to send the data to the database to be stored. After a new account has been created, the CRS will start the process of creating a new job record. This begins in the AccountManager class,

generating an ID for the job record. A JobRecord class is created which holds the job ID, the client's account ID and the description of the problem. After this is completed, the AccountManager class sends the new job record to the database to be stored.

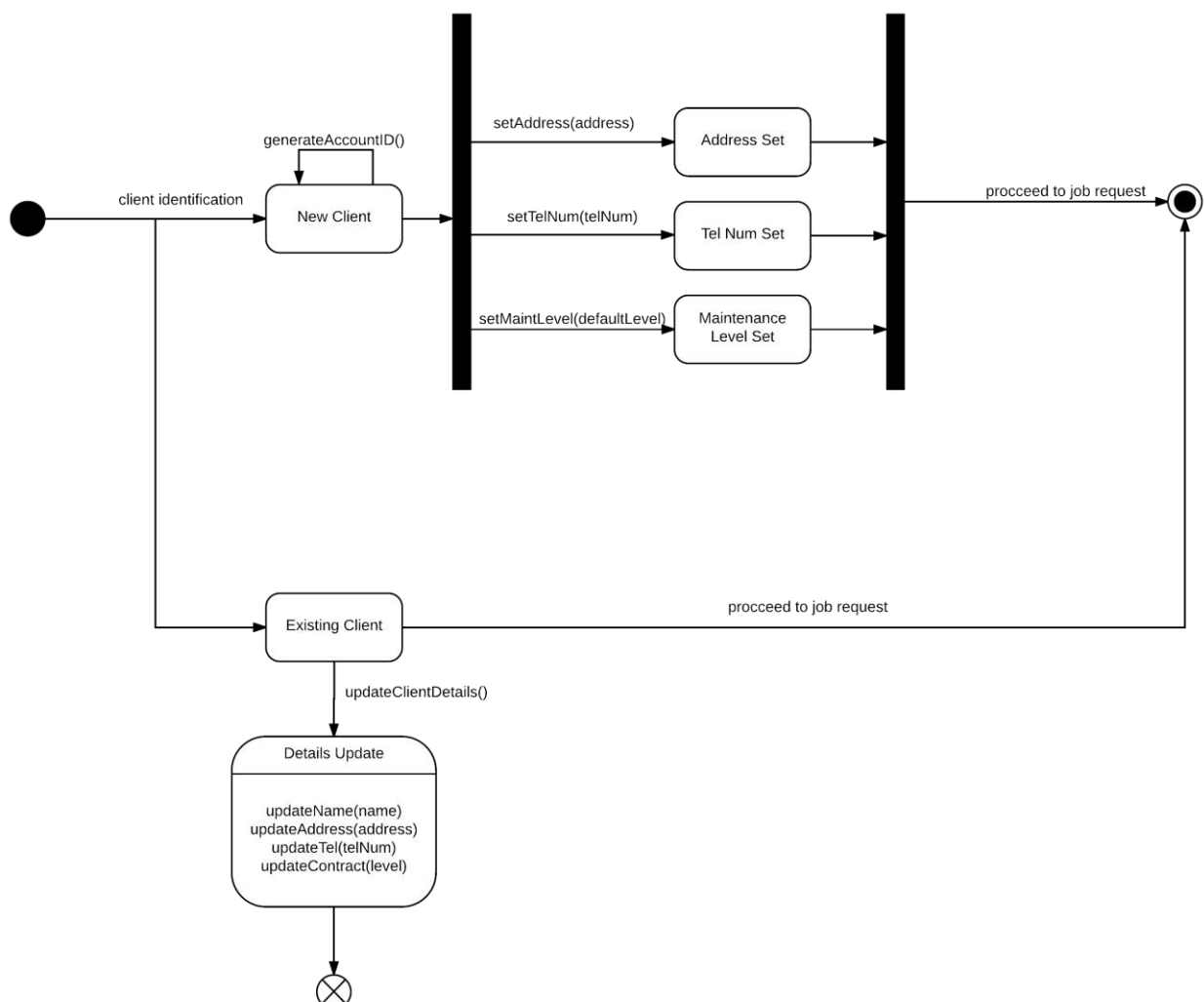


Update Client Details

This sequence diagram shows the process of a CRS modifying the client's account information. The CRS searches for an existing client using the client's accountID via the AccountManager, and will respond with a Boolean value. Once the client has been found,

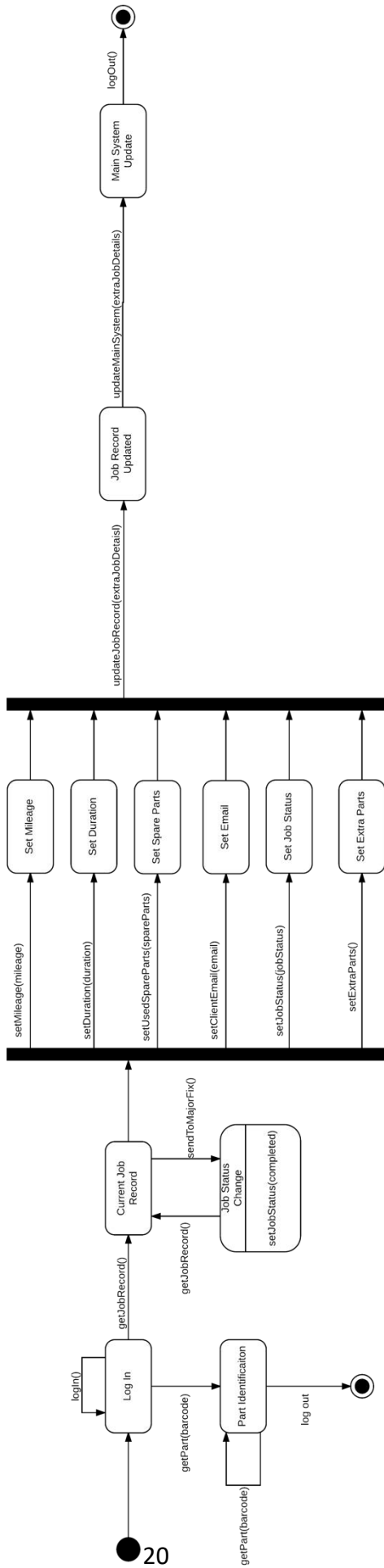
the CRS proceeds with updating their account details – this includes their name, telephone number, and address. These updates are sent to the database via the AccountManager.

ii. State diagrams



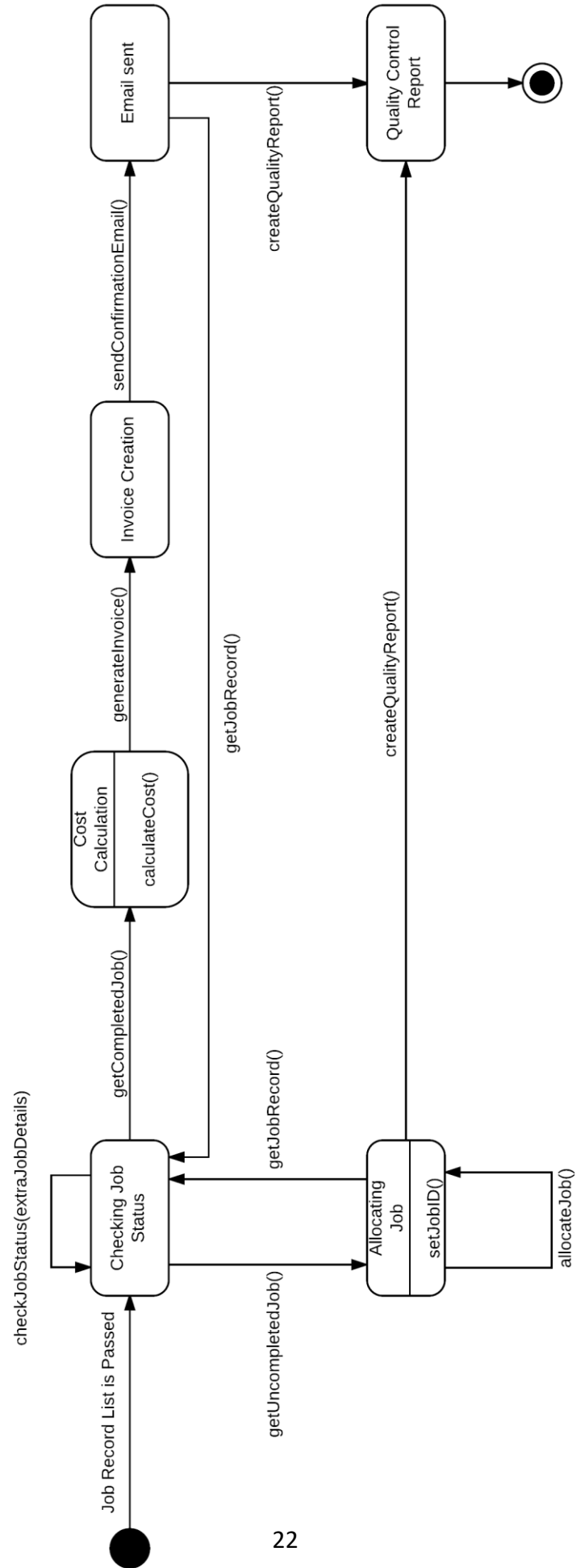
Account Manager State Diagram

Upon identifying the client, there are two different initial states for the system. In case of a new client state, an account ID is generated and the necessary account details are set (client's name, address and telephone number – the maintenance level is set to 0 by default). Once all account details are recorded the system can progress to record the client's service request. In case the client calling is a returning client, the system can either immediately proceed to a creating a new service request, or update the existing client's details and terminate after that.



Mobile Device Manager State Diagram Description

Either a member of the Engineers or the Warehouse Staff is provided with a mobile device and therefore can initiate the device's first log in state. If the log in is performed by a member of the warehouse staff, the mobile device is capable of identifying parts by their barcode and when necessary proceed to a log out state. On the other hand, if the log in is performed by an engineer, there are two possible scenarios as far as the mobile device is concerned. First of all, in case the job cannot be performed on-site, the job is immediately forwarded to MajorFix, the job status is marked as completed and the mobile system proceeds to the next job record. Otherwise, the extra job details of the current job record can be modified. From that point onwards, the system can set any one of the extra job details or all of them if needed. Additionally, the mobile device can update the job record and therefore update the main system with the job details for each job record. Finally, a log out state can be performed by both Engineers and Warehouse Staff members.



Administration State Diagram Description

Once a job record is retrieved the first system state is to check for the job status of the given record. For completed jobs, the system proceeds to calculate the cost. Further, provided the cost is calculated, the system creates an invoice that contains the job details as well as the price to be paid. Moreover, a confirmation email is sent to the client for approval and a quality control report is created. In the case of the job status being marked as uncompleted or partially completed, the system proceeds to set a job id for that job, allocate it and finally create a quality control report. Once this is done the system can get another job record and the entire process can be repeated multiple times.