**SE Engineering Project Notes:**

**System administrator:** act as super user to edit any data in the system

* Attributes:
  + employee surname: String
  + First name: String
  + Unique 5-digit ID: String

Operations:

+addUser(user: User) : void

+ editUser (user :User) : void

+deleteUser (user :User) : void

+changePassword (user: User) : void

* System users can use all other modules
* Administrator should be able to query the system: findUser (username: String /: registedUser)

**3.1 (Authentication)**

**Actor:** System admin

**Use case:** SystemAdmin logins to system

**Precondition:** System admin will be allocated an elevated authentication mechanism.

**Normal flow:**

1. System requests user enter their ID & Password
2. User enters their ID & Password
3. System validates the entered ID & Password and logs user into system

**Post condition:** User is logged into system and able to use all other modules

**Alternative flows:** At step 3 invalid ID & Password

3i) System detects that ID & Password is invalid

3ii) System displays error message saying invalid user name or password.

3iii) System error message offers a renewed invitation to log in again.

3iiii) exit flow and return to step 3

2)

**Use case:** Query System

**Actor:** SystemAdmin

**Pre-condition:** SystemAdmin logins correctly with correct ID & Password

**Normal flows:**

1. SystemAdmin request to query system to get list of active users and details
2. System returns users and their details

**Alternative flow at step 2:**

2i) System Admin in able to add, edit and delete directly from the list

2ii) When deleting a customer account confirmation shall be required

2iii) System admin confirms change in list

2iiii) Rejoin normal flow at step 2

**Post condition:** System user is viewing up to date list of system

**3.3 Vehicle Record (A)**

**Actor:** User

**Post-condition:** User is logged

**Use case:** Query system to get a list of vehicles with customer details and next booking date

Normal flow:

1. User query’s system
2. System returns a list of vehicles with customer details and next booking dates

**Post condition:** User views details

**Alternative flow:**

1i) User query’s system based on partial full registration number/manufacturer

1ii) Rejoin normal flow at step 2

**3.5 Parts (A)**

**Actor:** System Admin / User – day – to day

**Use case:** Manage parts

**Pre-condition:** User is logged in

**Normal flow:**

1. User queries system to get a list of parts used to repair a vehicle along with vehicle and customer details
2. System returns details with option to add delete or edit parts directly from the list
3. Confirmation of change pops up
4. System confirms and updates list with part withdraws and part additions for new stock items

**Post condition:** User views up to date parts system

**Alternative flow at step 2:**

2i) User requests to

2ii) System requests name, costs of part, description, and relevant ID number for part

2iii) User confirms change

2iiii) System confirms addition of part with installation date and warranty end date

2iiiii) Return to step 4

**Use case:** Manage past and future booking dates

**Actor:** day to day user/ admin

**Pre-condition:** User is logged in system

**Normal flow:**

1. User searches vehicle by registration number
2. System returns information of vehicles registration number
3. User queries system to get either a list of past or future booking dates
4. User requests to view future booking dates
5. Future/Up to date booking dates are displayed to the user

**Alternative flow at step 4:**

4i) User requests to delete future booking date

4ii) System requests user to confirm cancellation

4iii) User confirms cancellation

4iiii) Return to step 5

Post condition: Booking dates are displayed to the user