ECS506U Software Engineering
Group Project 2017

Problem Definition

Version 1.0

23 December 2017

Prof. Martin Neil

TABLE OF CONTENTS

1 Introduction

1. The aim of the 2017 software engineering group project (referred to as simply *the project* in the remainder of this document) is for your group to create and supply a stand-alone Java based implementation of a "Garage Management Information System (GM-SIS)".
2. It is important to note that the special requirements of the project are such that there is no point in students attempting to cheat by copying systems and websites online. The special requirements force groups to design their own solid object oriented code that they fully understand.
3. The requirements are organised into:
   - Functional requirements (Section 2)
   - Data requirements (Section 3)
   - Non-functional requirements (Section 4)
4. We have designed the requirements in such a way that you must apply solid object-oriented design principles in order to complete the project successfully.
5. The functional and data requirements for the system are final and as prescribed and thus not open to elucidation, negotiation or refinement, except at your own initiative.
6. You should read this document in conjunction with "Software Engineering Group Project 2017: Guidelines".

2 Functional requirements

The GM-SIS must provide the following seven modules:

- Authentication
- Customer account
- Vehicle records
- Diagnostic and Repair bookings
- Parts record
- Specialist Repairs
- Scheduled maintenance bookings

Each module, apart from the authentication module, will need to be developed separately and meet the following functional requirements:

1. Present a graphical user interface to the user to allow them to carry out all operations.
2. Allow records and data to be added/edited and deleted.
3. Correctly implement the specific data requirements associated with the module.

4. Save and retrieve records to/from a database table(s).
5. Execute specific operations on the data associated with the module.
6. Provide a method of user authentication.

The GM-SIS as a whole will be integrated by combining all of these independently developed modules together and will meet the following functional requirements:

1. Present a single GUI for the system as a whole.
2. Deliver a single authentication mechanism.
3. Preserve the data integrity of the system by ensuring that changes made in one module are updated in all others (e.g. deleting a vehicle record will delete the bookings associated with that record).

3 Module data requirements

The data requirements for each module are set out below. They delineate those data requirements specific to the module in question and those that belong elsewhere but which a given module must interact with.

Because the system is tightly integrated students are advised to understand requirements from ALL modules in order to best implement those relating to their module.

The authentication module can be developed by the group as a whole. There are six other modules specified in priority order. Groups with four members will do modules marked with an (A), those of size five will do those marked with (A) and (B) and those of size six will do (A), (B) and (C).

The delivered system should come with five customers already set up and with their associated data loaded. There should be sufficient data in the system for all modules to test each requirement thoroughly. A test plan and results will be required for each module.

3.1 Authentication (A)

1. A GM-SIS system administrator will be allocated an elevated authentication mechanism to allow them to setup and administer new and existing system users and to act as a super-user to edit any data in the system.
2. Records of all system users will be stored in the system.
3. There are two types of system users: system administrators and day-to-day users. Each will be identified by their employee surname, firstname and unique 5-digit identification number (id).
4. Garage customers are NOT users of the system.

5. A system user will logon using their id and a password. The password will be allocated by the system administrator
6. Only the system administrator can add, edit or delete system users and change passwords.
7. All system users can use all other modules in GM-SIS once they have logged in using their correct id and password.
8. Failure to login correctly will result in the system offering a renewed invitation to login again.
9. Checks shall be made to ensure that system users and ids are unique.
10. The administrator should be able to query the system to get a list of active users and their details. They should be able to add, edit and delete directly from this list.

3.2 Customer account (A)

1. A customer is a member of the public, who owns or cares for one or more vehicles, and who is a customer of the garage. There are two types of customers: businesses and private individuals.
2. Each customer account must store/include the following data: full name, address, postcode, phone, and email address.
3. One or more bookings may be made by the customer ("scheduled maintenance" and "diagnostic and repair") against one vehicle at a time.
4. Associated with each customer will be one or more bills for completed non-warranty diagnoses and repairs and scheduled maintenance bookings.
5. Each booking will have a bill (total cost) for the customer to settle. This is the "account" and the system must determine whether it is settled (paid) or outstanding (unpaid). Customers do not settle accounts for vehicles with valid warranties.
6. The user should be able to query the system to get a list of active customers with contact details. They should be able to add, edit and delete directly from this list.
7. For each customer, the system user should be able to access lists of past and future booking dates, vehicles and parts used per vehicle etc.
8. When deleting a customer account a confirmation shall be required.
9. At least 10 customer accounts must be created and integrated with data from other modules.
10. The user should be able to search for a customer by partial or full surname and firstname or vehicle registration number (number plate).

3.3 Vehicle record (A)

1. A vehicle is classified as a car, van, or truck.
2. A vehicle will either be under warranty or not. A warranty means that should a vehicle require diagnosis and repair, the warranty company pays for this and not the customer.

3. If a vehicle is under warranty the name and address of the warranty company and the date of expiry of the warranty must be recorded against the vehicle.
4. The details of each vehicle must be recorded by the garage: registration number, model, make, engine size, fuel type, colour and MoT renewal date.
5. Date from last service and the current mileage must be recorded.
6. Associated with each vehicle will be lists of parts used, past and future booking dates, and the total cost per booking (warranty and non-warranty). This must tally against the customer account.
7. A list of 20 standard vehicle template, such as Honda Civic 1.6 Litre SE, or Ford Focus 1.2 Litre, must be created and made available for quick selection when a new vehicle is being booked in, including model, make, engine size and fuel type.
8. The user should be able to query the system to get a list of vehicles with customer details and next booking date. They should be able to add, edit and delete directly from this list.
9. For each vehicle, the system user should be able to access lists of past and future booking dates, vehicles and parts used per vehicle etc.
10. When deleting a vehicle a confirmation shall be required.
11. At least 15 vehicle records must be created and integrated with data from other modules.
12. The user should be able to search for a vehicle based on partial or full registration number (number plate) or manufacturer.

3.4 Diagnosis and Repair booking (A)

1. There are two types of bookings: “diagnosis and repair” and “scheduled maintenance”. Each is handled differently but they share common features. Diagnosis refers to the act of finding a reported fault in a vehicle and repair refers to the act of fixing that fault. The fixing of a fault may require a repair to a part on the vehicle or the ordering and installation of a new, working, part.
2. Repairs are carried out by a single mechanic (a class of employee of the garage) whose hours spent on the repair are recorded. Each mechanic has a fixed hourly costing rate.
3. Each repair booking is of variable time duration.
4. The opening hours of the garage are Mon-Fri 9am to 5.30pm and Sat 9am-12 noon.
5. The garage is not open on public and bank holidays.
6. Appointments can only be made in the future.
7. Current mileage of the vehicle must be recorded during the diagnosis and repair booking.
8. The system user should be able to query the system to get a list of bookings with customer and vehicle details and next booking date. They should be able to add, edit and delete bookings directly from this list.
9. For each vehicle, the system user should be able to access lists of past and future booking dates, customer name and type of booking

10. When deleting a booking a confirmation shall be required.
11. It will be necessary to easily view and edit bookings such that lists of bookings can be viewed and edited on an hourly, daily and monthly basis.
12. At least 10 past bookings and 10 future bookings must be created and integrated with data from other modules.
13. The user should be able to search for a booking by partial or full vehicle registration number, vehicle manufacturer or customer surname and firstname.

3.5 Parts (A)

1. Each vehicle may require a new replacement part to affect the repair (if needed).
2. Each repair will involve zero, one or more parts.
3. Each part will have a name, a description and relevant id number (E.g. spark plugs, prop shaft, handbrake cable etc.)
4. The system must track the stock levels of parts currently available in the garage. Assume a limited list of ten distinct parts that can be installed on any vehicle (of whatever model and type).
5. The system must be able to track part withdrawals for repairs and part additions for new stock items.
6. The date the part was installed in a vehicle must be listed and the date the warranty for the item runs out (Each part has one year warranty from the day of installation).
7. The individual costs of parts must be recorded and used to calculate the customer's bill.
8. The system user should be able to query the system to get a list of parts used to repair a vehicle along with the vehicle and customer details. They should be able to add, edit and delete parts directly from this list.
9. For each vehicle, the system user should be able to access lists of past and future booking dates, customer name and type of booking
10. When deleting a booking a confirmation shall be required.
11. At least 20 stock items must be created and integrated with data from other modules. Records of at least 5 new stock deliveries and 10 stock withdrawals to repair vehicles must be shown. This data must be integrated with data from other modules.
12. The user should be able to search for a part used by partial or full vehicle registration number or customer surname and firstname.

3.6 Specialist Repairs (B)

1. Some repairs need to be sub-contracted to Specialist Repair Centres (SPC). This can involve sending the vehicle to the SPC or sending a part to be repaired or re-conditioned to that SPC.
2. Specialist repairs can be done as part of a single diagnostic and repair booking.

3. If it is a part that is being sent to an SPC the part undergoing specialist repair will have a name, a description and relevant id number (E.g. spark plugs, prop shaft, handbrake cable etc.)
4. If it is a vehicle that is being sent to an SPC the vehicle details must be recorded.
5. The garage uses ten SPCs, exclusively. The expected delivery date for the vehicle or part being sent to the SPC must be recorded along with the expected return date.
6. The name, address, phone and email address must be recorded for each SPC.
7. The system must allow system administrator's to add, edit and delete SPCs.
8. The system must be able to create a summary list of the number of parts and vehicles at SPCs outstanding (not yet returned).
9. The individual costs of parts and vehicle repairs sent to SPCs must be recorded and used to calculate the customer's bill.
10. Cost of SPC repairs and the cost of regular repair and diagnostics must be included in the same bill so that the customer needs to settle one bill only.
11. The system user should be able to query the system to get a list of specialist repairs used on a vehicle along with the vehicle and specialist repair details. They should be able to add, edit and delete parts directly from this list.
12. For each specialist repairer, the system user should be able to show lists of vehicles, and customer details per vehicle, sent to them.
13. When deleting a booking a confirmation shall be required.
14. At least 10 past specialist repair bookings and 10 future specialist repair bookings must be created and integrated with data from other modules.
15. The user should be able to search for a specialist repair by partial or full vehicle registration number or customer surname and firstname.

3.7 Scheduled Maintenance bookings (C)

1. There are two types of bookings: "diagnosis and repair" and "scheduled maintenance". Each is handled differently but they share common features. Scheduled maintenance refers to the act of servicing or inspecting a vehicle either to prevent faults arising in future or to ensure the vehicle is roadworthy.
2. There are two types of scheduled maintenance. The first is a service and the second is the MoT (Ministry of Transport) test.
3. A service is carried out according to the vehicle manufacturer's schedule, which is either usage or time based.
4. A usage based schedule stipulates the maximum number of miles the vehicle can be used between services, typically it is between 10000 and 12000 miles.
5. A time-based schedule stipulates the maximum elapsed calendar time between services, such as a year or two years.
6. Scheduled maintenance has a fixed cost per vehicle. For a MOT test the cost is £30 and for a service it is £150.
7. A vehicle can fail an MOT test and this must be recorded. The reasons for failure must be recorded.
8. Each scheduled maintenance booking is of fixed time duration.

9. The opening hours of the garage are Mon-Fri 9am to 5.30pm and Sat 9am-12 noon.
10. The garage is not open on public and bank holidays.
11. Appointments can only be made in the future.
12. Current mileage of the vehicle must be recorded during the diagnosis and repair booking.
13. It will be necessary to easily view and edit bookings such that lists of bookings can be viewed and edited on an hourly, daily and monthly basis.
14. At least 10 past bookings and 10 future bookings must be created and integrated with data from other modules.
15. The user should be able to search for a booking based on partial or full vehicle registration number, vehicle manufacturer or customer surname and firstname.

4 Non-functional requirements

In addition to correct implementation of the functional requirements, your project submission will be assessed against the following non-functional requirements:

- *Ease of installation*: This is the most critical non-functional requirement. You are expected to learn how to 'package up' your code in such a way that it can be executed with a click/invocation of the application from within the file system. Ideally this could be a jar, a batch or an exe file.
- *Portable*: Your application must run on any PC with an appropriate JVM and other necessary software installed on it.
- *Ease of use*: This is judged from the perspective of an end user. The elegance/attractiveness of the interface is also considered here.
- *Reliability/robustness*: The extent to which the code runs without failing. Some reasonable 'stress testing' must be performed.
- *Efficiency (speed):* System must response within acceptable time.
- *Efficiency (memory):* This will be judged on the amount of memory used when running, and the size of the application files.
- *Easy to Integrate:* Be developed in such a way that it will easily integrate any module with the other modules in the system.
- *Verisimilitude:* Be developed with sufficiently rich test data to able to demonstrate all functionality required.
- *Valid input data:* All input data is validated to ensure it meets requirements and constraints
- *Persistence:* All data input to the system should be persistent i.e. permanently stored for retrieval and use after the system has been closed down.