

UG Stage 2 Team Project (IN2018) module
2021/2022

All BSc and MSci students in the second year of their
studies, excluding students on Professional Pathway

Developing a Software Product

The GARage IT System (GARITS)

Student's Brief

Lecturer(s):

Dr Ilir Gashi

Centre for Software Reliability (CSR)

Department of Computer Science

Room: A401a Ext.: 0273, E-mail: Ilir.Gashi.1@city.ac.uk

Dr Vladimir Stankovic

Centre for Software Reliability (CSR)

Department of Computer Science

Room: A304j Ext.: 3079, E-mail: Vladimir.Stankovic.1@city.ac.uk

Authors: Ilir Gashi, Peter Mellor, Peter Popov, and Vladimir Stankovic.

Feb. 2022

Contents

| | | |
|------------|--|-----------|
| 1. | INTRODUCTION | 3 |
| 1.1 | PURPOSE AND SCOPE | 3 |
| 1.2 | LEARNING OBJECTIVES | 3 |
| 1.3 | PREREQUISITES | 3 |
| 1.4 | RECOMMENDED READING | 3 |
| 2. | ORGANISATION | 4 |
| 2.1 | TEAMS | 4 |
| 2.2 | THE "CUSTOMER" | 5 |
| 2.3 | THE CONSULTANT | 5 |
| 2.4 | COMMUNICATION | 6 |
| 3. | PROJECT TIMETABLE..... | 6 |
| 4. | PROJECT ASSESSMENT | 8 |
| 5. | SUBMISSION OF DELIVERABLES..... | 9 |
| 6. | ASSESSMENT CRITERIA FOR DELIVERABLES | 9 |
| 6.1 | GENERAL CRITERIA FOR DOCUMENTS..... | 9 |
| 6.2 | REQUIREMENTS SPECIFICATION AND SYSTEM DESIGN (DUE: WEEK 5, 40% OF TOTAL PROJECT MARKS) 10 | 10 |
| 6.3 | WORKING PRODUCT (DUE: WEEK 11 AND 12, 50% OF TOTAL PROJECT MARKS)..... | 10 |
| 6.4 | INDIVIDUAL FINAL REPORT (DUE: WEEK 12, 10% OF TOTAL PROJECT MARKS)..... | 11 |
| 7. | ORIGINALITY OF WORK..... | 11 |
| 8. | GARITS REQUIREMENTS | 12 |
| 8.1 | INITIAL REQUIREMENTS | 12 |
| 8.1.1 | <i>The Garage</i> | 12 |
| 8.1.2 | <i>Overview of the existing system</i> | 12 |
| 8.1.3 | <i>Detailed description of the existing system</i> | 13 |
| | Reception | 13 |
| | Garage bays | 13 |
| | Spare parts store | 14 |
| 8.1.4 | <i>Problems with the existing system</i> | 14 |
| 8.2 | GARITS REQUIREMENTS | 15 |
| 9. | DOCUMENTS PROVIDED BY CUSTOMER..... | 17 |
| 9.1 | CUSTOMER CARDS..... | 17 |
| 9.2 | JOB SHEET | 18 |
| 9.3 | MOT REMINDER | 19 |
| 9.4 | INVOICE..... | 20 |
| 9.5 | STOCK LEDGER | 21 |
| 9.6 | PARTS ORDER | 21 |
| 9.7 | SPARE PARTS / STOCK LEVEL REPORT | 22 |
| 10. | TEAM MEMBERSHIP..... | 23 |

1. Introduction

1.1 Purpose and Scope

This document is the Student's Brief about the Team Project module (IN2018). It describes the overall organisation of the project and the activities that each team are supposed to undertake. It presents project timetable, defines the module assessment, and details what each team and each student must deliver at each stage of the project. The document contains team membership, too. Also, the document includes initial requirements about the case-study that will be used throughout the project.

All students on the BSc (all routes) and MSci (all routes) degree programmes in the Computer Science department undertake the team project in their second year of the UG studies. The only exception to this are the students on the Professional Pathway route (they attend IN2030 Work-Based Project instead). Team Project module involves analysis, design, implementation and testing of a software system to meet a given set of requirements. A purpose of the project is to give the students an appreciation of the challenges and difficulties of working with peers in a team to satisfy the requirements of a fictitious commercial customer. Another purpose is for them to use software engineering methods to solve a realistically complex problem (that is, one that cannot be solved by an individual student in the time available), using the knowledge and skills gained in Stage 1 (e.g. IN1007 Programming in Java modules, etc.) as well as in Stage 2 (e.g. IN2013 Object-Oriented Analysis and Design). Yet another purpose, equally important as the ones above, is enhancement of transferable skills including teamwork, negotiation and interpersonal skills.

The module runs in Term 2. Week numbers used in the module are counted from the first teaching week of Term 2 and extend for 11 consecutive weeks until the end of the term (including the Reading/Project week in the middle). Some coursework deadlines are scheduled for week 12. The rationale is to allow the students to maximise their success.

Although this document outlines the tasks that each team must perform, it does not mandate any particular method of planning – this is left for the students to choose and follow. Also, although it is assumed that teams will deploy and use the UML and other methods formally taught during their courses, the teams are expected to exhibit a degree of initiative and to seek and apply solutions they have not already been taught. All teams will undertake the project based on the *same case-study*, but must work independently!

1.2 Learning Objectives

The learning objectives are to demonstrate the ability to plan, organise and conduct a software development project of realistic complexity, to document the software development with UML, and deliver a software package within schedule, and of adequate quality to meet customer requirements.

The ability to work together, and work effectively, with the other team members, to plan realistically, and to meet deadlines are important factors affecting success.

Specifically, apart from enhancing the relevant technical skills, the students will gain experience of the following:

- forming a working team, establishing common goals and resolving misunderstandings and conflicts;
- effectively communicating with a customer;
- planning time and effort and allocating tasks to individuals;
- identifying and resolving risks and problems;
- establishing and using document filing and configuration standards;
- working to a life-cycle with defined deliverables.

1.3 Prerequisites

Students are expected to draw upon material taught and experience gained in the Stage 1, as well as Stage 2 courses taught in parallel with Team Project. The following modules are particularly relevant:

- IN1007 – Programming in Java (Stage 1)
- IN1010 – Databases and Web Development (Stage 1)¹
- IN2013 – Object Oriented Analysis and Design (Stage 2). This is a **co-requisite** module.

1.4 Recommended reading

For most of the UML tasks especially relevant reading are as follows:

- Jim Arlow, Ila Neustadt, UML 2.0 and the Unified Process: Practical Object-Oriented Analysis and Design, Addison-Wesley, 2nd edition, July 2005, 592 pages, ISBN 0321321278

¹ From 2021-22 academic year, the module has been replaced by 15-credits "IN1013 Databases" module.

- M. Fowler, *UML Distilled: a brief guide to the standard object modelling language*, 3rd Edition, Addison-Wesley, 2003
- G. Booch et al, *Object-Oriented Analysis and Design with Applications*, 3rd Ed., Addison-Wesley, 2007
- S. Bennett et al, *Object-Oriented Systems Analysis and Design Using UML*, 4th Ed., McGraw Hill, 2010
- B. McLaughlin et al, *Head First Object-Oriented Analysis and Design: A Brain Friendly Approach to OOAD*, O'Reilly, 2006
- Cay, S. Horstmann, *Object-Oriented Design & Patterns*. Hoboken, NJ, Wiley. 2006
 - This text is useful especially to learn about design patterns, but also as a practical guide that links UML models with programming in Java.

For supplementing, and advancing, your Java knowledge and skills the following is useful:

- C. Horstmann, *Big Java*, 7th edition, "Relational Databases" Chapter, John Wiley and Sons.
 - <http://www.horstmann.com/bigjava.html>

For furthering your knowledge and skills in Human Computer Interaction, the following is recommended:

- D. Benyon, *Designing Interactive Systems*, 4 Edition, Pearson, 2019

Relevant web resources are as follows:

- UML Specification
 - <http://www.omg.org/spec/UML/2.5.1/>
- Object Management Group, UML Resource Page
 - <http://www.uml.org/>
- Practical UML: A Hands-On Introduction for Developers
 - <http://edn.embarcadero.com/article/31863>
- Allen Holub's Quick UML Reference
 - <http://www.holub.com/goodies/uml/>
- Java tutorial on JDBC
 - <http://download.oracle.com/javase/tutorial/jdbc/>

Please see also the reading materials suggested in the related modules above. Further information about the relevant literature for the module can be found at: <http://readinglists.city.ac.uk/index.html> . See also the Library Guide for the CS department - [Computing Lib Guides](#).

2. Organisation

2.1 Teams

Before the official start of the module, the students are assigned to teams of 6 (+/- 1) members, all of whom will be on one of the department's undergraduate degree courses. The size of the teams was decided based on many years of successful running of the module – the team size is chosen to allow a **degree of redundancy** so that in case a team member withdraws, or lacks commitment, this does not necessarily lead to the team failure. The rationale behind the pre-assignment of students to teams is to ensure that teams have a **balance of talent, skills and expertise** and become aware that in their future careers one must learn to establish good working relations with "strangers" as well as acquaintances and friends (see sections "1.1 Purpose and Scope", and "1.2 Learning Objectives" for further explanations). Allocation to the teams will be done **based on the marks achieved** in Stage 1 (IN1007), and only *secondarily* based on coursework marks in IN2013. The students ranked in the top, middle and bottom of the cohort are **equally** represented in each team. This ensures as fair as possible team membership for *all* students.

The students may **not** change the team once assigned (no expulsions, no poaching etc.).

A **team consultant**, who is a member of the Team Project Management Team (TPMT), will be allocated to each team. The consultant role(s) are described below – see section "2.3 The Consultant".

Each team will work independently of others on the same case-study. The internal organisation of each team, planning the project, defining a management structure, carrying out the work (technical or organisational) etc., is the *joint responsibility of all the members* of the team, and nobody else. Each team member is expected to adopt two roles, with associated job titles, within the team and to take responsibility for those tasks falling within their remit.

Teams are encouraged to follow a *compacted* version of *waterfall process* for software development in line with the team project deliverables. But, other software development processes can be adopted (even if only in part) – this is up to the teams to decide. Also, the teams **must** use **UML** for requirements specification, analysis and design.

However, they are free to choose their programming language, database platform, operating system and machine type (provided that *they have the right kit, e.g. laptops etc., themselves* in order to demonstrate the software to the TPMT during the “Demo” day). Traditionally, most teams implement their software using Java, for reasons of familiarity, and support availability. However, teams are free to use other programming languages, e.g., C++, Python, JavaScript, C# or any of the languages supported by the .NET platform, etc.

Each team member will adopt **two** roles from the following list:

- **project manager** : responsible for team planning, co-ordination, and risk management; he/she acts as primary interface to the team consultant and the customer (NB this person **must** contribute technical work too!);
- **deputy project manager** : second in command, and responsible for documentation, reports and standards (NB: this person must contribute technical work too!);
- **system analyst** : responsible for elicitation of customer requirements and requirements analysis and specification;
- **designer** : responsible for the system design process;
- **programmer** : responsible for implementing the system;
- **tester** : responsible for writing a test plan, testing the system and its components and recording the outcomes.

Each team member will have one **primary role** and a **secondary role**, e.g., project manager/systems analyst or designer/programmer. The roles must be appropriately spread. This will ensure that essential roles within the team are covered when members are absent. There can only be one project manager role. A student cannot be both project manager and deputy, of course. An example of a typical configuration of roles within a team might be:

- Sarah: project manager/programmer
- Bob: deputy project manager/systems analyst
- Jim: designer/tester
- Ashraf: designer/programmer
- Tamara: systems analyst/programmer
- Niles: designer/tester

Despite the roles assigned to each member, it is strongly suggested that **all students be actively involved in, and contribute to, the work throughout the project** (though not necessarily with same level of involvement and contribution in each phase).

Teams must operate autonomously. **All** students should be present at **all briefing and Q&A sessions**. Members of staff will be available to answer questions at every Q&A session.

Each team must keep a file of minutes of all their meetings, and a record of all tasks completed, documents (or parts of documents) delivered including the “rough”, draft versions, etc. (as either hard-copies or e-copies). This content should be stored in the **Project Binder**, which is also referred to as **Project Manual**.

In addition, **each student must** keep an **Individual Diary** of their own involvement in the project. An **individual plan must be drawn up** at the beginning of the project showing the **expected hours** to be put in during the project, broken down by tasks. Then, a record should be kept of the **actual effort** (in a time unit of choice, usually hours) spent and a reason should be given for any substantial discrepancy. In addition to helping students contribute to the project in an effective way, this will be an input to the Individual Report, which is a piece of the formal, summative assessment due in the end of the project (see below).

Both the Project Binder and the Individual Diaries must be shown to the respective team consultant on demand. *In any case intermediate versions must be submitted to the respective consultant at the end of each software development phase!* The team consultant will assess the individual contribution of each member of the team at the end of the project, based upon these documents as well as other evidence e.g., attendance and involvement at the meetings with the consultant, the Individual Reports (see Section 6.4 below), etc.

2.2 The “Customer”

Your project is to deliver a software package to meet the requirements of a “customer”. Staff will play the role of the customer representatives: Mr. Lancaster, and other members of staff in his company. The students will have to discover the full requirements by interviewing them. The initial statement of requirements is included in this document (Section 8). **You are expected to have read carefully and analysed the text in Section 8 and the material in Section 9 (if any) before going for interviews with the representatives of the customer!**

The customer will evaluate some of the documents the teams deliver, including the final product. The other documents will be evaluated by the team consultant (or other members of the TPMT).

2.3 The Consultant

The **primary role** of the team consultant is to monitor progress of the team, and help resolve difficulties and risks communicated to him or her.

Please note that not all consultants are equally versed in every part of the Team Project tasks. Team consultants **are not meant to act as technical consultants** to the teams. This is, partly, so that TPMT remain fair to all teams. If any technical advice is given by a consultant, it is the team's decision to use it or not. Technical questions, instead, are meant to be directed to the Customer/lecturer(s).

2.4 Communication

Most communication during the Team Project will be done by e-mail or Moodle. Therefore, the students must ensure that they are able to send and receive e-mail and check their e-mail inboxes, and Moodle messages, regularly.

There will be at least 4 categories of messages exchanged during the project:

1. *Internal messages between members of the team.* How to organise internal communication is up to the team members. Some may be deemed strictly private. Others will need to be filed in the Project Binder to record progress or setbacks.
2. *Messages between the team and the customer.* These may have "contractual" significance, and teams should use a procedure for filing them for future reference. Most of them will need to be included in the Project Binder.
3. *Messages between the team and the consultant.* These should be dealt with separately from customer messages. (They will normally be kept confidential from the customer.).
4. *Messages from the Team Project module leader* will deal with general administrative matters.

The teams are also encouraged to use Moodle for internal team communication (see <http://moodle.city.ac.uk>). Please note that each team is provided with a private Discussion Forum on Moodle. The team members, the respective consultant, and the lecturer(s) have access to a given forum.

Meetings are also a vital means of communication, and essential for success. There are, at least, the following meeting types: i) internal team meetings and ii) team consultant meetings. Teams should hold the former type of meetings regularly at appropriate frequency, and arrange to see their consultant – the latter type of meetings – *weekly* online via Teams or Zoom. Please note that team consultants are very busy (they are lecturers, PhD students, or full-time teaching assistants), and will need sufficient notice of appointments. A consultant may only be able to grant appointments at certain times/days, but will see each of the teams for **up to 30 minutes every week**. For consultant meetings, the most effective way is to have a dedicated day of the week, and time, agreed in advance between the team and the consultant. The consultant meetings are **mandatory**!

3. Project Timetable

The project will extend throughout the whole of the teaching Term 2 (31 January – 15 April), including the Reading, i.e. Project, week. Some coursework will be due in **week 12, after the teaching finishes**, to give you sufficient time to succeed. The deadlines have been agreed and optimised with the lecturers of all other Stage 2, Term 2 modules, in order to give you appropriate time to succeed in all modules. Moreover, all coursework deadlines have been published in the beginning of the academic year. The timetable of activities and deliverables (outputs) is given in Table 1. The contact hours and face-to-face assessments are on the official timetable too.

| Week | Briefings/Q&As/Deadlines | Activities | Deliverables |
|------|---|---|--------------|
| 1 | 1 February: Briefing on module organisation (an extra lecture in week 1) 3 February: Briefing on Requirements Specification. Guest lecture on team-working NB: There are two 2-hour lectures (Briefings) in Week 1. This is for students' benefit – to help you succeed in the project by having an additional Briefing early in the module. | (Re-)Read Student's Brief in detail. Company name and logo. Role and responsibility definition. Set-up Project Binder. Start Individual Diary, devise Individual plan of effort in the project. | None |
| 1, 2 | 3 February: Interviews 10 February: Database modelling recap | Interviews with customer - further requirements elicitation. Students must attend the respective interview. See the schedule on Moodle | None |

| | | | |
|------|--|--|--|
| 1–5 | <p>17 February: Briefing on System Design and Q&A session</p> <p>24 February: Java GUI recap and Version Control Guest Lecture/Tutorial</p> <p>3 March: Briefing on System Implementation</p> <p>Sun. 6 Mar, 5pm (end of week 5): Submission of: “Requirements Specification and System Design” document, and Project Binder and Individual Diaries.</p> | <p>Specifying requirements and modelling system design.</p> <p>Project Binder and Individual Diaries maintenance.</p> | <p>“Requirements Specification and System Design” document – formal, summative assessment to be submitted on Moodle.</p> <p>Individual Diaries and Team Project Binder (to your consultant)</p> |
| 6–12 | <p>17 March: General Feedback on the first deliverable, and Q&A session.</p> <p>24 March: Briefing on Database Connectivity programming (JDBC)</p> <p>31 March: Guest talk on Agile Methodology and Test-Driven development, and Q&A session.</p> <p>7 April: Preparation for Software Demo, and Q&A session.</p> <p>TBC: Week ending 17 April: Demonstration of the software in operation – summative assessment, and if the pandemic situation allows the Showcase event (Thur. 14 April, Whole day); Submission of Implementation Reports (Sun. 17 April, 5pm);</p> <p>TBC: Week ending 24 April: Submission of Individual Reports; (Sun. 24 April, 5pm);</p> <p>TBC: Mon. 25 April, 5pm Final versions of Project Binder and Individual Diaries, and any final evidence about individual contribution by any of the team members!</p> | <p>System Implementation and Testing.</p> <p>Attendance to the Demo and Showcase event.</p> <p>Preparation and submission of Implementation Reports.</p> <p>Preparation and Submission of the Individual Reports</p> <p>Project Binder and Individual Diaries maintenance, and submission.</p> | <p>Formal, summative assessments:</p> <ol style="list-style-type: none"> 1. Working system demonstration (“Demo”) 2. Implementation Report 3. Individual Report <p>Individual diaries and Team Project Binder (to your consultant)</p> |

Table 1: Team Project timetable of briefings, activities and deliverables

Formal Briefing materials (slides) will be prepared and presented in class during the module. The aims of these materials are to introduce what is expected at each stage in the project and to give *some* guidance on how to tackle it. The briefings are **not a replacement for formal software engineering or OO analysis/design/programming lectures – most, if not all, of the material needed for Team Project will have been covered in other modules!** The timetable for the **briefing sessions** and Q&A sessions are as follows:

- Week 1 (1 February – Tue!) – Introduce Team Project and its structure, learning objectives, assessments, etc.
- Week 1 (3 February) – What is a requirements specification? Guest lecture on effective team-working.
- Week 2 (10 February) – Database modelling recap.
- Week 3 (17 February) – What is a system design? Q&A session, too.
- Week 4 (24 February) – Version control and Java GUI recap
- Week 5 (3 March) – What is software implementation?
- Week 7 (17 March) – General Feedback on the first deliverable. Q&A session.
- Week 8 (24 March) – Java JDBC programming
- Week 9 (31 March) – Guest talk on Agile Methodology and Test-Driven development, and Q&A session.
- Week 10 (7 April) – Preparation for Software Demo, and Q&A session.
- Week 11 (14 April) – Software Demo assessment, and if the pandemic situation allows Showcase event [whole day event; no lecture this week]

The project will proceed according to the following stages:

1. *Team formation and project definition*

During the first class, the lecturer will present the purpose, organisation and timetable of the team project. The students will be informed of their membership in pre-selected teams. ***This is non-negotiable!*** Teams will decide upon a company name and logo to appear on all of their correspondence and deliverables. Each team must set up their Project Binder. Team members' roles and responsibilities will be documented in this binder. The team must decide when these roles are most prominent within the project. Each team member is expected to deliver 100% commitment to the team according to the roles assigned.

2. *Requirements Specification*

The customer requirements for the project will be presented. There will be opportunities for questions. Teams must prepare for, and attend the interviews with the representatives of the fictitious Customer. UML will be used at this stage to model the system. A *requirements specification* will be developed.

3. *System design*

The software architecture of the system will be designed by choosing appropriate decomposition of the system into sub-systems, which will be identified, and their designed interaction documented. This will model the new system as it is intended to be upon implementation. Again, UML will be used here. A system *design* will be developed.

One document, containing both requirements specification (see above) and system design, is to be submitted as the first summative assessment deliverable of the module.

4. *Implementation*

The system will be coded in its entirety and tested to establish whether it meets the requirements. Also, Implementation Report will be prepared which should include:

- description of how the system must be compiled (if applicable), deployed, and executed
- testing report/plan for unit/use-case testing, and
- commented source code.

Using the UML diagrams related to the implementation phase is recommended, but is not mandatory.

The working system will be evaluated via demonstration of the final product ("Demo"), which mainly consists of checking if the functional requirements are met. Usability of the software will be evaluated too. Implementation Reports are included in this phase too.

5. *Individual report*

Each student will reflect individually on what has been achieved. He/she will report on lessons learnt and problems experienced, and describe their own, and other members', contribution to the project.

4. Project Assessment

The following is the table of all coursework components to be delivered by each team (and individual), together with the respective proportion of marks:

| Item | Description | Week | Mark (%) |
|------|--|------|----------|
| 1 | Requirements Specification and System Design document | 5 | 40 |
| 2 | System Demonstration ("Demo"), and Implementation Report | 11 | 50 |
| 3 | Individual Report | 12 | 10 |
| | Total | | 100 |

Table 2: Team Project Deliverables and Marking scheme

Any team/individual that fail(s) to meet a deadline, and have not been granted extension after following the EC process, as specified by the relevant department's and the university's regulations, will be awarded ZERO marks.

The reasons for the marks awarded will be communicated to the teams in a timely manner so as not to unduly penalise their performance in the next stage. For instance, a team that submits a poor system design and so receives a low mark will be given feedback about what errors/omissions were made so that they have a chance of producing a correct implementation. There will, however, be **no re-assessment of revised documents** to take account of the

feedback provided after the marking has taken place. Producing a document revised in line with the received feedback is not required.

Please note that feedback will be given throughout the module, but especially after each phase, e.g. in the class (general feedback), during Q&A sessions, during lecturer's Office Hours, in private communication with students etc.

Where members of a team find the contribution of a member to be unsatisfactory, the team is encouraged to resolve the issue internally by negotiation and constructive discussion. If the problem cannot be resolved internally then the team should ask the consultant to intervene. Depending on the circumstances, the consultant might decide to involve the lecturer(s)/module leader(s) in some cases.

In addition, the consultant might during the project, based on clear evidence from a variety of sources (see below), communicate to her/his respective teams and individual team members any concerns known to him/her. The **students are, however, responsible** for informing the consultant of all relevant affairs so that the consultant can form a full and correct view.

VERY IMPORTANT

At the **end** of the Team Project the **final mark allocated to each individual team member will be adjusted in line with the individual's contribution**, despite it being based on the overall marks achieved by the team. In this way, we guarantee fairness to all: non-attendance, breached agreements, lack of contribution etc., can be recognised and the individual final marks awarded to the individuals after adjusting them in line with their respective relative contribution. All adjustments will be justified and backed up by objective evidence such as:

- depositions by team co-workers;
- attendance and involvement in the team's meetings, and the meetings with the team consultant;
- notes from the team consultancy meetings;
- depositions in the Individual Reports;
- depositions in the Individual Diaries and Project Binder;
- depositions from various forms of communication among team members, e.g. via Moodle, email, social media;
- Etc.

Any amendments to individual marks are at the consultant's discretion (and in agreement with the module leader), and are based on provided evidence by all students by a fixed, announced in advance, deadline.

This procedure guarantees a **fair marking approach for each team member!**

5. Submission of Deliverables

For summative assessment, there are two documents² to be delivered by each team (Requirements Specification and System Design deliverable, and Implementation Report deliverable), and one to be delivered personally by each student (Individual Report deliverable).

All summative assessment documents must be submitted **electronically via Moodle by the respective deadline**.

The dates for submitting deliverables will appear on Moodle in due time and are tentatively listed in Table 1. They must be strictly adhered to.

Deliverables will be marked, and feedback given **no later than three, or four, weeks** of the submission, as specified by the relevant regulations for project, coursework-only modules. Please see <https://staffhub.city.ac.uk/policies/student-and-academic-services-policies/assessment-and-feedback-policy> and <https://www.city.ac.uk/about/governance/policies/student-policies-and-regulations>.

In addition, after each of the two phases – Analysis & Design, and Implementation – each team must submit the Project Binder, and each student must submit the respective Individual Diary to the team consultant. Project Binder and Individual Diaries are not marked, but they do serve an important purpose: they are used as an evidence when assessing each student's individual contribution (see e.g. Section 4).

6. Assessment Criteria for Deliverables

6.1 General criteria for documents

Some written documents submitted for marking will be assessed using the general criteria covering presentation and style. In addition to this, each deliverable will be assessed according to the criteria covering its content as listed below. Fairness of the marking will be ensured by the same section of all deliverables being marked by the same marker(s).

² The teams will also need to prepare the summative assessment that is not a document: Demo of the working product.

For Requirements Specification & System Design document, and Individual Reports, 10% of the overall mark will be awarded for General Criteria.

General Criteria:

- Appropriate title, page numbering and version control (2)
- Introduction, including Purpose & Scope, of the document; and description of the old system (if one applies) (2)
- Use of language appropriate to audience (consultant vs. customer), and Spelling and Grammar (2)
- Clear layout and structure (2)
- Appropriate use of graphics and diagrams (2)

Total: 10%

6.2 Requirements Specification and System Design (Due: Week 5, 40% of total project marks)

This deliverable will be assessed against the following particular criteria:

- Requirements Specification
 - Full use case diagram(s) defining collections of use cases and their interactions with actors covering the whole functionality of the new system; (20)
 - Use case specifications for **10 key use-cases**, with main and alternative course of actions made clear, actors defined, pre- and post-conditions stated and all interactions specified; (15)
 - Indexed list of all use cases prioritised according to users' priorities and impact of projected risks during development (i.e., time and budget problems). This should help drive the evolution of the design and implementation tasks. (5)
- System Design
 - Fully refined and correct Design class diagram(s) showing Entity, Boundary (i.e. GUI) and Control classes, associations (including roles and navigability), cardinalities, methods (i.e. operations) and attributes. A complete set of operations should be specified including parameter lists, return types, visibility, set and get operations/methods, and constructors. Also, a complete set of attributes including types and default values must be provided.
The class diagram needs to include classes from the implementation domains (e.g. DB connectivity). Packages should be used to show the system architecture and the interfaces (and the respective implementation classes) between the sub-systems. (20)
 - ER diagram, and relational database schema (specified to the 3rd Normal form) represented with a complete set of DDL statements (CREATE TABLE statements). Also, a representative set of Data Manipulation Language (DML) statements (2 SELECTs, 2 INSERTs, 2 UPDATEs and 2 DELETEs) must be provided i) adhering to the team's DB schema and ii) with meaningful values. Also, all necessary SQL DML statements which are needed to create **2** non-trivial reports pertaining to the case-study must be provided.
 - Class diagram is not the same as database schema. In any case, for this phase the teams are expected to use Relational Database Management Server (RDBMS), not an Object-Relational, or an Object-Oriented DBMS, or a NewSQL data store. The database chosen should offer transactional support. Also, the specific DB product assumed for the DB design **must** be stated. (20)
 - GUI designs. **Design** the visual appearance of the GUI forms (screenshots); **map** these to the boundary classes shown in the class diagram; and **show** how the users will **navigate** through the GUI, i.e. through the chosen set of GUI menus/forms etc. (10)

Total: 90%. The remaining 10% of the marks for this deliverable will be awarded for the General Criteria.

The target audience for this deliverable is the customer *and* the team consultant. It should provide explanatory material in layman's terms (i.e., use cases) complemented by technical specifications represented in a structured fashion (e.g. class diagram).

6.3 Working Product (Due: Week 11 and 12, 50% of total project marks)

Accompanying the final software system should be an Implementation Report describing the implemented software, problems encountered in implementing the initial design, solutions/fixes, and the results of testing.

A copy of the fully commented source code, or in the case of code generated automatically the appropriate files used by the code generating software, should be submitted in the dedicated submission area on Moodle, as a .zip file. **Failure to supply the source code will result in Zero marks being awarded for the whole of the Working Product stage.**

The working product will be assessed using the following criteria:

- Working product Demo
 - Completeness and quality of the product in relation to the requirements (75)
 - Ease of use and consistency of the GUI (5)
- Implementation Report
 - Software architecture/Compilation/Run-time components (8)
 - Testing plans and reports (12)

Teams using code generation environments (e.g. NetBeans, Eclipse, IntelliJ IDEA, Visual Studio etc.) should describe the structure of their implementation by referring to the following:

- Forms and their properties
- Controls and dialog boxes, menus etc., and their properties
- Database tables
- Coded modules

During the working product "Demo", a checklist will be used by the staff to award marks for the presence of features in the product which are stated in the requirements, and also the quality of implementation of those features. Functionality will be assessed using **two sets of scenarios**, which describe the expected functionality of the product. These scenarios, which will be executed with the completed product after its deployment on a target machine, will be made available to the teams as follows:

- i) The first set of scenarios will be given to the teams **at least two weeks before the final demo**. The teams must use this set to prepare for the final demo. Before the start of their final demo the teams are expected to have cleared the database used by the application and to have executed the entire first set of scenarios. **At the start of the "Demo" assessment the database used by the application should contain the data generated by executing the scenarios included in the first set and nothing else.**
- ii) At the demonstration each team will be presented with another set of scenarios, similar to those included in the first set distributed in advance, and asked to execute them with their product.

The marker of the Demo will ask the demonstrator(s)/team to show features in accordance with the requirements.

The software demonstration ("Demo"), and the Showcase event, is planned to take place in week 11. The details about the Demo, and the Showcase event if one is to be held, (including the schedule, etc.) will be **confirmed about two weeks before the demonstration**.

The target audience for the product is the customer, except for the Implementation Report, which is for the consultant.

6.4 Individual Final Report (Due: Week 12, 10% of total project marks)

The target audience for the Individual Report is the consultant. Students ought to use the teams' Project Binder and the respective Individual Diary as inputs when writing the Individual Report. The individual report will be assessed using the following criteria:

- Table of the effort expended in each phase against estimated effort. In case of (substantial) discrepancy an explanation must be provided. Details need to be provided about *own* contribution to the team deliverables. (25)
- Description of how the project went and statement of any problems *personally encountered* and how they could be avoided in future. (15)
- Description of how the team worked together, and providing evidence-based assessment about *the contribution of the other members* to the team deliverables. (15)
- Lessons learnt. (20)
- Statement of what you would do differently in future. (15)

Total: 90%. The remaining 10% of the marks for this deliverable will be awarded for General Criteria.

7. Originality of Work

Although the teams are starting from the same statement of requirements, each team is expected to work independently and to tackle the problem in their own way. Obviously, there will be overall similarities in the

deliverables from different teams, and there will be plenty of opportunities to see what other teams are doing, either in the class/Q&A sessions or in private discussions between the members of different teams.

However, copying of (parts of) documents or software of one team by another is, of course, not allowed, and will be treated as plagiarism of coursework!

If copying is detected³, all students/teams involved will be given **zero** marks for the copied deliverables.

8. GARITS Requirements

Your company has been invited to develop a “GARage IT System (GARITS)” for a branch of Quick Fix Fitters plc - a national chain of franchised garages. Quick Fix Fitters are worried that their competitors are increasing market share because they offer a better service and generate a higher rate of repeat service. Head office believes this is because the competition use computerised systems. Quick Fix Fitters want you to design and test an IT system for a single garage franchise. Based on this experience they may or may not commit themselves to going national. They may decide to offer your company repeat contracts for other franchises, if you do a good job.

Your consultant has won the contract from Quick Fix Fitters head office to assess their requirements and produce an implementation of *one system for one garage*. Your team has been tasked with doing the job. Your consultant asks you to set up an appointment with the franchisee of the Ashford branch of Quick Fix Fitters garage – Glynne Lancaster. Mr Lancaster has made a first stab at defining the system (he has even named it – GARITS), which he thinks is needed for his garage.

8.1. Initial Requirements

The Garage IT System (GARITS) will keep records of customers and their vehicles. It will send reminders to customers whose vehicles are due for MoT test in the following week. When a vehicle is booked in for servicing, GARITS will generate a job sheet for the mechanic, describing the work to be done. When the job is completed, the time spent and the prices of the spare parts fitted will be added up and an invoice will be available to the customer.

GARITS must be easy to use, but may run on any hardware/software platform that it is convenient to use. Also, Quick Fix Fitters Head Office want a system that can be easily reused in other franchised garages up and down the country.

To get a complete picture of the requirements you must interview Mr Glynne Lancaster, the franchisee at Ashford branch of Quick Fix Fitters.

8.1.1. The Garage

The business contains a reception, a repair shop, a yard and a parts store. It deals with cars and light-goods vehicles (vans). It employs following people: 2 receptionists, 8 mechanics, 1 foreperson, and 3 storekeepers. Much of the trade for the repair shop consists of supplying MoT certificates, but it also does regular services (10,000 mile etc.) and accident repairs, etc. It has a regular clientele consisting of members of the public and business customers. It also picks up some casual trade. The repair shop has 2 MoT inspection bays and 8 repair bays that each take one car or one van.

They are a major dealership for “Fjord Motors” – a new Norwegian “no-gimmicks” car manufacturer who have built a new plant up North. It also sells new and second-hand vehicles from other manufacturers.

The parts store sells parts to other garages and the public. They stock major parts for all the main manufacturers and vehicle types but stock *all* parts for Fjord Motors.

8.1.2. Overview of the *existing* system

Currently the customers can book their vehicles in for a service, MoT or repair. This is dealt with by reception, who can take bookings by phone or email. Repair or MoT bays are allocated to customers using a “wipeable” marker board located between the reception and repair shop. A check is made of the customer card if necessary (in case they owe the garage money). If all is ok, a job sheet is prepared for the repair shop detailing the problem, etc. For account customers, reception can despatch a mechanic to pick up the vehicle. Other customers deliver their vehicles for a service.

³ Checks will be made! For example, via automatic comparison of source code submissions, etc.

The garage also gets drop-ins where no prior booking has been made. The vehicle waits in the yard until a repair bay is available. The customer is contacted with a price estimate and asked to confirm the service. When non-account customers come to collect their vehicles they are presented with an invoice listing, the job done and price, which they must **pay on the spot**.

Any parts needed for a job are withdrawn from the store. If the store does not have the part, then it will be ordered and picked up from another garage or the manufacturer. Orders are emailed ahead of a driver who goes out to pick up the part. If there is a problem in getting a part, the customer will be contacted and the vehicle will stay in the yard until repaired.

8.1.3. Detailed description of the *existing* system

Reception

Checks in vehicles for repair, answers queries, types invoices and letters, maintains card index of *Account Holders* customers, and their vehicles. Lots of email/telephone work and the reception is always hectic.

Customer Cards

Card index maintained in filing cabinet - one card per customer. Cards also used to provide details for invoicing on a very old computer in Mr Lancaster's office.

Job Sheets

A job sheet is issued upon check-in and is completed in duplicate. One copy of the job sheet gets filed in the job filing cabinet, and the other is stuck on the whiteboard using a magnet for pick up by Evans, the foreperson. When, and if, Evans (or another mechanic) returns his/her copy, the details are used to prepare the invoice. However, sometimes the cost of a job may be high, and Evans passes the copy back to reception so the customer can be informed.

Job allocation - The White board

Jobs are booked in for repair or MoT bays as necessary by pinning them on the whiteboard against bays. It is up to Evans to decide which mechanic to allocate upon inspection of the job sheet.

Invoices

Prepared from completed job sheets. Evans passes through the partially completed job sheet; reception contact the store to get prices for parts, then complete the job sheet and contact the customer by phone (or a mechanic delivers the vehicle).

A duplicate copy of the invoice is filled in - one for the company's records and one for the customer. Some customers pay on the spot when collecting their vehicles, but others (*Account Holders*) have the invoice posted to them the following (working) day, and are asked to pay in 30 days' time. Payments can be made using cash or credit/debit card. Invoices are calculated as follows:

- Labour rate = £105 per hour for mechanics, apart from Evans, who is worth £125 per hour. This excludes VAT.
- Parts are bought at trade price and 30% is marked up before they are sold to customers.
- VAT is added to the total bill when composing the invoice (i.e., to both labour *and* parts).

One of the job sheet copies is given to the customer with the invoice. The other is held on file. Once the customer pays, stamp "paid" is put on the invoice.

Garage bays

A receptionist issues a job sheet when checking a vehicle in. This identifies the owner, the vehicle, its make, model and registration number, and what needs to be done (if this is known at the time). The job sheet is put up on the whiteboard denoting which repair bay it should be allocated. Evans examines the job and selects a free mechanic to

have a look at it. The mechanic is then in possession of the job sheet until the job is finished or the customer needs to be contacted.

After the job is completed the mechanic fills in the details: what was done, time taken, spares used, etc. (On many occasions, no-one on the repair shop floor is willing to do this, and the delays cause a lot of frustration for Mr Lancaster). The sheet if/when complete (!) is returned to Mr Lancaster.

Usually, an estimate is given before the job is started. It may be discovered during the job that it will cost more than was thought. Also, a vehicle may be brought in for a routine MoT, which it fails, and (expensive) repairs may be required before it will pass. In either case, Reception is informed and the job sheet passed to them.

Spare parts store

This department is managed by Ms Eva Kournikova, the Senior Storekeeper.

Stock Control

Stock control is enforced through the ledger that contains a list of current parts held (alphabetically). Ledger page is split into columns with part name, manufacturer, vehicle type, year(s) and number of parts listed. Pencil is used to update the current parts total as stock moves in and out (the ledger is almost transparent in places where a rubber has been repeatedly applied over the years).

Stock Reception

In addition to operating the stocks on behalf of the garage, the store also deals with members of the public who buy parts for their own DIY repairs. The storekeepers also deal with telephone queries from customers about cost of parts as well as with customers coming to buy parts. Once a part is purchased, the ledger is updated and one of two invoices is presented to the customer (stamped as paid). The other is filed away for record keeping purposes.

Parts Ordering and Invoicing

At the end of the day Ms Kournikova reviews what parts have been sold and, if necessary, places replenishment orders for replacements with manufacturers. However, she holds back until she feels she can place a bulk order. She keeps the size of these bulk re-order quantities in her head. She is, however, able to list the stock-point at which an order must be placed and the amount to be ordered on paper, but cannot explain why it works so well.

If a part is not in stock, an order is sent by email to a supplier or the manufacturer and a driver sent ahead to collect and bring back the part. Suppliers/manufacturer send invoices for the order direct to the reception not to the store.

Reporting

Has to produce a monthly report to Mr Lancaster on inventory costs and levels. Currently this is not very easy and Kournikova hates doing it.

8.1.4. Problems with the existing system

Mr Lancaster is concerned about keeping the garage's Account Holders customers and also expanding the business by turning some of the casual customers into Account Holders.

The garage also has a number of corporate clients, small local firms who run fleets of vehicles, and Mr Lancaster is keen to expand that area of the business. He has been losing these business customers to other local garages and he suspects the paper-based procedures they currently use may be to blame. Reductions in job response times by creating an efficient computerised system might help here.

The garage wish to remind their customers when their next MoT is due, in the hope that they will come to Quick Fix Fitters for the check-up and any necessary repairs. They also hope that this will present a caring and efficient image to their corporate clients. It would be nice if they could do the same for the annual service of customer vehicles.

For MoT tests that are due in the following week, the garage would like to send a letter to the customer reminding them it is due and suggesting a test time. What does “in the following week” mean? Obviously, if the vehicle's MoT test is due on a Monday, there is no point in sending a reminder on the previous Friday. The reminder must be timed so that there are 5 *working days* between the reminder and the due date.

It is barely possible to remind customers about impending MoT tests with the existing system, even for corporate customers, since each of their vehicle's MoT test falls due on a different date. (The first MoT test on a new vehicle is due 3 years after the initial purchase/registration, and yearly on the same day/month thereafter.)

Corporate customers with large fleets of vehicles cause problems: the customer cards aren't big enough to list all the vehicles. (It is useful to know how many and of which make and model the garage is likely to get, as this assists spare parts ordering, etc.)

The current system of job sheets gets inconsistent very often, especially when a vehicle is left in the yard awaiting a part. A job can get accidentally raised again by another receptionist the next day, who will issue a new set of job sheets. As a result sometimes the customer receives two job sheets attached to the invoice or the reception staff spend ages sorting out the mess.

Ms Kournikova admits that she is getting older and her memory is failing. It has been noticed that parts shortages are occurring more frequently and that several mechanics are concurrently out on the road collecting parts. There must be a way of automating the parts re-ordering process.

The ledger is also getting out of synchronisation with the actual number of parts.

The number of customers who purchase spare parts is on a sharp decline because the store takes too long to locate the part, update the ledger, and fill out an invoice and process payment.

Fjord Cars have been very dissatisfied with the performance of the garage when servicing their cars. (The garage is subjected to an audit every quarter.) Fjord hinted they would like to see an automated inventory system (whatever that is).

8.2. GARITS Requirements

The new system should eliminate all problems with the existing system listed in 8.1.4, and thus allow for more efficient operation of the garage. In addition, the following features must be provided by GARITS:

- Different levels of access should be implemented with the following roles being implemented:
 - **Administrator.** This role should be the only one allowed to deal with the *user* accounts – adding new user account, removing/changing an existing account. Also backing up, and restoring, the database can only be performed by the administrator. The role cannot do anything that the other roles (see below) can do with GARITS. A default *admin* account shall be created on deploying the GARITS software.
 - **Franchisee.** This role will have a full access to the functionality related to running the garage (i.e. anything that is related to taking in new jobs, processing them, generating reports, dealing with the spare parts, etc.). This role can create, and alter, a customer account, and is the only role that can configure “pay late” option and set/alter a customer's discount plan, which can be one of the following:
 - a *fixed* discount, i.e. the same percentage of discount is given to the customer for each job and parts. The value of a discount of this type is calculated and deducted from the value of the job at the time of accepting the job;
 - a *variable* discount, i.e. the percentage of the discount is set and may vary between the different types of business, e.g. MoT, service types (individual tasks), spare parts. The value of a discount of this type is calculated and deducted from the value of the job at the time of accepting the job.
 - a *flexible* discount, i.e. the percentage of the discount depends on the value of the orders by the same customer accumulated within a calendar month. The value of the orders and the applicable discount should be calculated by GARITS at the end of the month (or on demand). The value of the discount is paid back to the client by deducting from the value of a future order.
 This role, however, cannot assume any privileges associated to the Administrator role.
 - **Foreperson.** This role can take in new jobs, add them to the list of pending jobs, allocate mechanic to a particular job, etc. In short, they can do everything that a mechanic and receptionist can do (see the next two bullet points).
 - **Mechanic.** They should be allowed to pick a job from the list of pending jobs and alter the job's status by filling in details about their work on a vehicle as follows (see Section 9 for an example):
 - specifying the individual operations performed (selecting from a list of tasks predefined for the garage). For each operation, i.e. task, the ‘default’ duration will be specified (time it normally takes a mechanic to

complete the operation). The mechanics, however, should be able to alter this duration (no checks are expected from GARITS whether the new value specified by the mechanics is accurate). The system should record the mechanic who has undertaken the work (i.e. the one typing in the effort expended) and use his/her hourly rate to calculate the labour cost later, when an invoice for the service is generated;

- specifying the spare parts used in the service (repair) of the vehicle. For this the mechanics should be able to access the database of spare parts and choose the ones used in the service. At this point the system should reduce the stock of the spares by the number of units used.
- **Receptionist.** They shall be able to do the following
 - Take in new jobs (typically this will be done together with the Foreperson who will provide the details of the tasks to be completed).
 - Receptionist can create a customer record, if needed.
 - They should also be able to monitor the progress made with jobs (including searching for a specific job by car number, name of the customer, etc.).
 - Produce an invoice, once all tasks listed in the job sheet have been completed.
 - Do *anything related to stock control*, i.e. parts ordering, including searching the stock of spare parts (at least by model of the car, or type of the spare part) and reporting about any aspect related to spare parts. Thus, in the new system, the role of Receptionist can carry all the activities the storeroom staff can do in the existing system (including all activities of (Senior) Storekeeper).
- Mr Lancaster needs the following types of reports to efficiently run the business:
 - number of vehicles booked in on a monthly basis, overall and per job type (MoT/Annual service/repair), and type of customer (casual or Account Holder)
 - average time, and price, per job type (i.e. MoT/Annual service/repair). This should be done overall, and per job type (MoT, annual service, repair, etc.), and / or given mechanic
 - stock-level, i.e. spare parts, report (see below for further explanation)
 All types of reports must be both viewable and printable.
- Mr Lancaster is anxious that the new system be easy to use by everyone who needs to use it (but is unsure about who would need to do so), so appropriate GUI must be developed. He knows that convincing the mechanics to use the system may be difficult and would like some recommendations about what to do about this. He also wants a quality system - no cheap rubbish but no "Rolls Royce" either.
- The system should calculate and print invoices, and this would relieve reception staff of a lot of routine work (and could even cut down staff numbers!).
- The *late payments*, by Account Holders, should be detected automatically by the system and the Franchisee should be alerted about these. If the Franchisee is logged-in, the system will generate alerts, e.g. via pop-up messages, warning at regular intervals of (e.g. 15 minutes) until the Franchisee acknowledges the receipt of the alert.
- A stock control system is needed to find a part, price it up, and update the stock level.

Automated alert of missing/low-stock-level parts is required. For every spare part type, an alert (e.g. a pop-up window) should be generated when the respective stock level is below a threshold. The default threshold level is 10, but this ought to be configurable for each spare part type. In addition, a stock-level report (see "spare parts report" above, and Sec. 9. below) should be produced with a list of part description, manufacturer and re-order quantity etc. It should be possible to generate the report either weekly or on-demand.

It is important that an appropriate Concurrency Control mechanism is used in the system, so that database inconsistencies are precluded. For example, two, or more mechanics, when specifying the parts used for respective jobs (see the description of the Mechanics role above) should perceive, and leave the database with, consistent stock level; the correctly configured system should prevent the mechanics from choosing the very same instance of a spare part from the database. A similar situation occurs when the database stock levels are being updated (as a result of the new items arriving in the garage) in parallel with the database updates from the mechanics.

To fully understand all requirements you **must** interview Mr Lancaster. Before the interview, you must be clear with the content of this document and other material/information communicated to you regarding the Garage IT System (GARITS) requirements. The interview dates and times have been arranged already.

9. Documents provided by customer

9.1 Customer Cards

CUSTOMER/VEHICLE RECORD

Date: 25/10/2021
Name: Andrew J. Little
Address: 5 High St., Stratford, Essex
Post Code: CM1 7YY
Tel.: 0208-753-9828
Email: little@company.ltd

| Reg. No.: | Make: | Model: | Eng.Serial: | Chassis Number: | Colour: |
|-----------|-------|--------|-------------|-----------------|---------|
| EX68 ZJG | Ford | Focus | 819765237J | X3246751 | Pink |

CUSTOMER/VEHICLE RECORD

Date: 27/01/2022
Name: J. Hancock
Address: 2, Railway Cuttings, East Cheam, London
Post Code: EJ12 8GJ
Tel.: 0208-936-3756
Email: J.Hancock@gmail.com

| Reg. No.: | Make: | Model: | Eng.Serial: | Chassis Number: | Colour: |
|-----------|----------|----------|-------------|-----------------|---------|
| HU65 SAY | Vauxhall | Insignia | 12984352A | 57634257J | Puce |

CUSTOMER/VEHICLE RECORD

Date: 30/03/2020
Name: Matthew & Sons, Ltd.
Address: 17 Oxford Road, Croydon
Post Code: CR12 7AZ
Tel.: 0208-421-73591 Ext. 412 (J. Smith, Transport Manager)
Fax.: 0208-421-73533

| Reg. No.: | Make: | Model: | Eng.Serial: | Chassis Number: | Colour: |
|-----------|-------|--------|-------------|-----------------|---------|
| UZ63 YAS | Ford | Mondeo | 876912654 | 35287456 | Red |
| WR68 SAY | Ford | Mondeo | 769342653 | 76354248 | Red |
| LX61 HGF | Ford | Mondeo | 654396558 | 34527865 | Red |
| KN09 KLJ | Ford | Mondeo | 457863806 | 23517867 | Red |
| TT11 OPI | Ford | Mondeo | 567437965 | 34672876 | Red |

9.2 Job Sheet

Quick Fix Fitters,
19 High St.,
Ashford,
Kent CT16 8YY

JOB NO.: 83745

Vehicle Registration No.: DF65 POT
Make: Opel
Customer Name: J. Smith

Date Booked In: 18th October 2021
Model: Vectra Estate
Tel.: 0208-964-8093

Description of work required:

- 1) Replace exhaust
- 2) Strip head and replace worn valves

Estimated time: 4 hours

Description of work carried out (to be filled in on completion):

- 1) Exhaust replaced
- 2) Stripped head: found valves were worn - these replaced
- 3) Also checked grommet seals and found they needed replacing.

Actual time: 5.75 hours

Spare parts used (to be filled in on completion):

| Description | Part No. | Qty. |
|--------------|-------------|------|
| Exhaust pipe | X784/6352J | 1 |
| Head Gasket | Y76432-89T5 | 1 |
| Valves | 672351X/34K | 6 |
| Grommet Seal | X784/4563-2 | 3 |

Signature:

9.3 MoT Reminder

J. Smith,
27 Sainsbury Close,
Stratford,
Essex EJ6 5TJ

Quick Fix Fitters,
19 High St.,
Ashford,
Kent, CT16 8YY

18th January 2021

Dear Mr. Smith,

REMINDER - MoT TEST DUE

Vehicle Registration No.: DF65 POT Renewal Test Date: 25th January 2021

According to our records, the above vehicle is due to have its MoT certificate renewed on the date shown.

Account Holders customers such as yourself are assured of our prompt attention, and we hope that you will use our services on this occasion in order to have the necessary test carried out on your vehicle.

Yours sincerely,

G. Lancaster

9.4 Invoice

J. Smith,
27 Sainsbury Close,
Stratford,
Essex EJ6 5TJ

Quick Fix Fitters,
19 High St.,
Ashford,
Kent CT16 8YY

18th October 2021

Dear Mr. Smith,

INVOICE NO.: 197362

Vehicle Registration No.: DF65 POT
Make: Opel
Model: Vectra Estate

Description of work:

- 1) Replace exhaust
- 2) Strip head and replace worn valves
- 3) Replace grommet seals

| Item | Part No. | Unit Cost | Qty. | Cost (£) |
|--------------|-------------|-----------|------|----------|
| Exhaust pipe | X784/6352J | 57.50 | 1 | 57.50 |
| Head Gasket | Y76432-89T5 | 15.75 | 1 | 15.75 |
| Valves | 672351X/34K | 5.15 | 6 | 30.90 |
| Grommet Seal | X784/4563-2 | 0.90 | 3 | 2.70 |
| Labour | | 105.00 | 5.75 | 603.75 |
| Total | | | | 710.60 |
| VAT | | | | 142.12 |
| Grand Total | | | | 852.72 |

Thank you for your valued custom. We look forward to receiving your payment in due course.

Yours sincerely,

G. Lancaster

9.5 Stock Ledger

Stock Ledger

| <i>Part Name</i> | <i>Code</i> | <i>Manufacturer</i> | <i>Vehicle Type</i> | <i>Year(s)</i> | <i>Price</i> | <i>Stock level</i> |
|------------------|-------------|---------------------|---------------------|----------------|--------------|--------------------|
| Grommet | X66745877 | Fjord | Krapa | 2014-2021 | £1.90 | 32 |
| | D43-78 | Vauxhill | Ofcorsa | 2010-2021 | £1.30 | 12 |
| Water Pump | G457 | Fjord | Krapa | 2010-2014 | £56.70 | 6 |
| | H456-9UI | Volva | S34 | 2008-2015 | £124.34 | 2 |
| ... | | | | | | |

9.6 Parts Order

Quick Fix Fitters,
19 High St.,
Ashford,
Kent CT16 8YY

Date: 09th August 2021

Company: Fjord Distribution Ltd
Address: 25 The Causeway, Staines, Middlesex

Tel: 01784 407862
Fax: 01784 407863

| <u>Order Number</u> | <u>Description</u> | <u>Quantity</u> | <u>Price</u> |
|---------------------|--------------------|-----------------|--------------|
| X93456 | Engine Block | 3 | £ 34.56 |
| C4563 | Radiator | 4 | £145.00 |
| Total: | | | £179.56 |

Signed:

9.7 Spare Parts / Stock Level Report

Quick Fix Fitters,
19 High St.,
Ashford,
Kent CT16 8YY

Spare Parts / Stock Level Report

Report Period: 01/11/2021 – 30/11/2021

| Part Name | Code | Manufacturer | Vehicle Type | Year(s) | Price | Initial Stock level | Initial cost, £ | Used | Delivery | New Stock level | Stock cost, £ | Low level threshold |
|--------------|-----------|--------------|--------------|-----------|---------|---------------------|-----------------|------|----------|-----------------|---------------|---------------------|
| Grommet | X66745877 | Fjord | Krapa | 2011-2015 | £0.90 | 34 | 30.60 | 2 | 0 | 32 | 28.80 | 10 |
| | D43-78 | Vauxhill | Ofcorsa | 2010-2021 | £1.30 | 12 | 15.60 | 0 | 0 | 12 | 15.60 | 10 |
| Water Pump | G457 | Fjord | Krapa | 2010-2014 | £56.70 | 8 | 453.60 | 0 | 3 | 11 | 623.70 | 10 |
| | H456-9UI | Volva | S34 | 2008-2015 | £124.34 | 2 | 248.68 | 0 | 0 | 2 | 248.68 | 3 |
| Total | | | | | | | 748.48 | | | | 916.78 | |

Report Date: 10th December 2021

Senior Storekeeper:

Ms E. Kournikova

10. Team membership

| Team | First Name | Surname | Email ID | Consultant |
|------|------------|------------------|-------------------------------------|---------------------------|
| 1 | Aadil | Saiyad | Aadil.Saiyad@city.ac.uk | Cefa.Karabag.2@city.ac.uk |
| | Brigen | Ujkaj | Brigen.Ujkaj@city.ac.uk | |
| | Cameron | Sim | Cameron.Sim@city.ac.uk | |
| | Omar | Khodabaksh | Omar.Khodabaksh@city.ac.uk | |
| | Sourav | Singh | Sourav.Singh@city.ac.uk | |
| 2 | Adil | Wadud | Adil.Wadud@city.ac.uk | Cefa.Karabag.2@city.ac.uk |
| | Mohamad | Kanso | Mohamad.Kanso@city.ac.uk | |
| | Muhammad | Abdullah | Muhammad.Abdullah.3@city.ac.uk | |
| | Samiya | Ahmed | Samiya.Ahmed.2@city.ac.uk | |
| | Sirio | Bambino | Sirio.Bambino@city.ac.uk | |
| | Sumayyah | Islam | Sumayyah.Islam@city.ac.uk | |
| 3 | Andrei | Stan | Andrei.Stan@city.ac.uk | Cefa.Karabag.2@city.ac.uk |
| | Bilal | Bismillah | Bilal.Bismillah@city.ac.uk | |
| | Charles | Odili | Charles.Odili@city.ac.uk | |
| | Jakub | Dziarnowski | Jakub.Dziarnowski@city.ac.uk | |
| | Michal | Kaszubski | Michal.Kaszubski@city.ac.uk | |
| | Pavithira | Thavarajah | Pavithira.Thavarajah@city.ac.uk | |
| 4 | Ammaar | Murshid | Ammaar.Murshid@city.ac.uk | Cefa.Karabag.2@city.ac.uk |
| | Charles | Jacobs | Charles.Jacobs@city.ac.uk | |
| | Dhiresb | Hirani | Dhiresb.Hirani@city.ac.uk | |
| | Mudhir | Sharif | Mudhir.Sharif@city.ac.uk | |
| | Nayyaf | Alotaibi | Nayyaf.Alotaibi@city.ac.uk | |
| | Philippos | Giavridis | Philippos.Giavridis@city.ac.uk | |
| 5 | Abdalaziz | Alibrahim | Abdalaziz.Alibrahim@city.ac.uk | Jason.Jacques@city.ac.uk |
| | Andreas | Salcedo-Cespedes | Andreas.Salcedo-Cespedes@city.ac.uk | |
| | Hamza | Rehawi | Hamza.Rehawi@city.ac.uk | |
| | Mohammed | Saleem | Mohammed.Saleem@city.ac.uk | |
| | Muhammad | Rahman | Muhammad.Rahman.6@city.ac.uk | |
| | Siley | Malakbaba | Siley.Malakbaba@city.ac.uk | |
| 6 | Calum | Kellie-Smith | Calum.Kellie-Smith@city.ac.uk | Jason.Jacques@city.ac.uk |
| | Furkan | Alsancak | Furkan.Alsancak@city.ac.uk | |
| | Mahdi | Malkawi | Mahdi.Malkawi@city.ac.uk | |
| | Melvin | Fernandes | Melvin.Fernandes@city.ac.uk | |
| | Mohammed | Shabir | Mohammed.Shabir@city.ac.uk | |
| | Youssef | Abdelmoamen | Youssef.Abdelmoamen@city.ac.uk | |
| 7 | Dylan | Tuckey | Dylan.Tuckey@city.ac.uk | Jason.Jacques@city.ac.uk |
| | Erjon | Krasniqi | Erjon.Krasniqi@city.ac.uk | |
| | Hardik | Narotam | Hardik.Narotam@city.ac.uk | |
| | Ismacial | Aden | Ismacial.Aden@city.ac.uk | |
| | Mitchell | Massiah | Mitchell.Massiah@city.ac.uk | |
| | Ritharshan | Makenthiran | Ritharshan.Makenthiran@city.ac.uk | |
| 8 | James | Hamlyn | James.Hamlyn@city.ac.uk | Jason.Jacques@city.ac.uk |
| | John-Luca | Kutschera | John-Luca.Kutschera@city.ac.uk | |
| | Luis | Bedoya Castano | Luis.Bedoya-Castano@city.ac.uk | |
| | Maudud | Ahmed | Maudud.Ahmed@city.ac.uk | |
| | Mohammed | Rahman | Mohammed.Rahman.31@city.ac.uk | |
| | Tate | Apcher | Tate.Apcher@city.ac.uk | |
| 9 | Azhar | Sharif | Azhar.Sharif@city.ac.uk | Martin.Walter@city.ac.uk |

| | | | | |
|----|------------------|--------------------|--|--------------------------------------|
| | Fred | Alcantara | Fred.Alcantara@city.ac.uk | |
| | Humzah | Chowdry | Humzah.Chowdry@city.ac.uk | |
| | Mohammad | Iqbal | Mohammad.Iqbal.7@city.ac.uk | |
| | Patrik | Chrenko | Patrik.Chrenko@city.ac.uk | |
| | Sadman | Kibria | Sadman.Kibria@city.ac.uk | |
| 10 | Ahmad | Al-Homaid | Ahmad.Al-Homaid@city.ac.uk | Martin.Walter@city.ac.uk |
| | Jakub | Mastalerz | Jakub.Mastalerz@city.ac.uk | |
| | Maximillian | Ovesen | Maximillian.Ovesen@city.ac.uk | |
| | Rizwan | Mirza | Rizwan.Mirza@city.ac.uk | |
| | Tariq | Baluch | Tariq.Baluch@city.ac.uk | |
| | Yahye | Farah | Yahye.Farah@city.ac.uk | |
| 11 | Adil | Mughul | Adil.Mughul@city.ac.uk | Martin.Walter@city.ac.uk |
| | Ayesha | Kayani | Ayesha.Kayani@city.ac.uk | |
| | Meraz | Hossain | Meraz.Hossain@city.ac.uk | |
| | Nabil | Talukder | Nabil.Talukder@city.ac.uk | |
| | Nadim | Zaman | Nadim.Zaman@city.ac.uk | |
| | Nubbannie | Saani | Nubbannie.Saani@city.ac.uk | |
| 12 | Fahim | Bhuiyan | Fahim.Bhuiyan@city.ac.uk | Martin.Walter@city.ac.uk |
| | Marcel | Binek | Marcel.Binek@city.ac.uk | |
| | Mohammed | Rahman | Mohammed.Rahman.33@city.ac.uk | |
| | Paulius | Raginskas | Paulius.Raginskas@city.ac.uk | |
| | Umer | Faisal | Umer.Faisal@city.ac.uk | |
| | Waleed | Hassan | Waleed.Hassan.2@city.ac.uk | |
| 13 | Deepak | Singh | Deepak.Singh.2@city.ac.uk | Abigail.Parisaca-Vargas.1@city.ac.uk |
| | Jakub | Rozenek | Jakub.Rozenek@city.ac.uk | |
| | Muhammad | Ali | Muhammad.Ali.16@city.ac.uk | |
| | Saeed | Azizi | Saeed.Azizi@city.ac.uk | |
| | Timur | Mahmud | Timur.Mahmud@city.ac.uk | |
| | Youness | Said | Youness.Said@city.ac.uk | |
| 14 | Abdisamad | Siad | Abdisamad.Siad.2@city.ac.uk | Abigail.Parisaca-Vargas.1@city.ac.uk |
| | Adam | Kaczmariski | Adam.Kaczmariski@city.ac.uk | |
| | Adel | Zaky | Adel.Zaky@city.ac.uk | |
| | Alexandru-Cosmin | Prisacariu | Alexandru-Cosmin.Prisacariu@city.ac.uk | |
| | Shah | Ahmed | Shah.Ahmed@city.ac.uk | |
| | Zainab | Mayet | Zainab.Mayet@city.ac.uk | |
| 15 | Kristof | Mayer | Kristof.Mayer@city.ac.uk | Abigail.Parisaca-Vargas.1@city.ac.uk |
| | Maksymilian | Szejko | Maksymilian.Szejko@city.ac.uk | |
| | Pedro | Tamen-Oguike | Pedro.Tamen-Oguike@city.ac.uk | |
| | Raj | Parmar | Raj.Parmar@city.ac.uk | |
| | Said | Khan | Said.Khan@city.ac.uk | |
| | Xinyan | Luo | Xinyan.Luo@city.ac.uk | |
| 16 | Jhonny | Medrano Chuquirima | Jhonny.Medrano-Chuquirima@city.ac.uk | Abigail.Parisaca-Vargas.1@city.ac.uk |
| | Lewis | Brooks | Lewis.Brooks@city.ac.uk | |
| | Paulius | Kancleris | Paulius.Kancleris@city.ac.uk | |
| | Umar | Nizam | Umar.Nizam@city.ac.uk | |
| | Zainab | Karim | Zainab.Karim@city.ac.uk | |
| | Zihad | Wadud Ansari | Zihad.Wadud-Ansari@city.ac.uk | |
| 17 | Abelash | Rangarajah | Abelash.Rangarajah@city.ac.uk | Marius.Zicius.2@city.ac.uk |
| | Ali | Mohammed | Ali.Mohammed.2@city.ac.uk | |
| | Aymen | Ahmed | Aymen.Ahmed@city.ac.uk | |
| | Gleb | Trusov | Gleb.Trusov@city.ac.uk | |

| | | | | |
|----|-----------------|---------------|-------------------------------------|------------------------------|
| | Husnain | Waseem | Husnain.Waseem@city.ac.uk | |
| | Omar | Mohamed | Omar.Mohamed.2@city.ac.uk | |
| 18 | Andreea-Mihaela | Cafingiu | Andreea-Mihaela.Cafingiu@city.ac.uk | Marius.Zicius.2@city.ac.uk |
| | Fabiha | Khan | Fabiha.Khan@city.ac.uk | |
| | Jack | Murphy | Jack.Murphy.2@city.ac.uk | |
| | Mohamed | Ga'Al | Mohamed.GaAl.2@city.ac.uk | |
| | Mohammad | Baboa | Mohammad.Baboa@city.ac.uk | |
| | Vadims | Pasinicevs | Vadims.Pasinicevs@city.ac.uk | |
| 19 | Abd-Al-Samad | Syed | Abd-Al-Samad.Syed@city.ac.uk | Marius.Zicius.2@city.ac.uk |
| | Andin | Mikushnica | Andin.Mikushnica@city.ac.uk | |
| | Luqmaan | Ali | Luqmaan.Ali@city.ac.uk | |
| | Saajidah | Mohamed | Saajidah.Mohamed@city.ac.uk | |
| | Sebastian | Chan | Sebastian.Chan@city.ac.uk | |
| 20 | Ahsan | Latif | Ahsan.Latif@city.ac.uk | Marius.Zicius.2@city.ac.uk |
| | Haatim | Rahman | Haatim.Rahman@city.ac.uk | |
| | Iqram | Hussain | Iqram.Hussain@city.ac.uk | |
| | Ismaeel | Aamir | Ismaeel.Aamir@city.ac.uk | |
| | Muhsin | Rahman | Muhsin.Rahman@city.ac.uk | |
| | Shadman | Efti | Shadman.Efti@city.ac.uk | |
| 21 | Akos | Szente | Akos.Szente@city.ac.uk | Warren.Fernando.2@city.ac.uk |
| | Estella | Himona | Estella.Himona@city.ac.uk | |
| | Farasat | Khan | Farasat.Khan@city.ac.uk | |
| | Neil | Bouguerra | Neil.Bouguerra@city.ac.uk | |
| | Poojah | Poobalasingam | Poojah.Poobalasingam@city.ac.uk | |
| | Zahid | Hussain | Zahid.Hussain.2@city.ac.uk | |
| 22 | Anson | Lam | Anson.Lam@city.ac.uk | Warren.Fernando.2@city.ac.uk |
| | Daniel | Kostadinov | Daniel.Kostadinov@city.ac.uk | |
| | Mahin | Ahmed | Mahin.Ahmed.3@city.ac.uk | |
| | Muhammed | Hasan | Muhammed.Hasan@city.ac.uk | |
| | Oliver | Van Til | Oliver.Van-Til@city.ac.uk | |
| | Tameeca | Naidoo | Tameeca.Naidoo@city.ac.uk | |
| 23 | Ahmed | Omer | Ahmed.Omer@city.ac.uk | Warren.Fernando.2@city.ac.uk |
| | Jakub | Dudziuk | Jakub.Dudziuk@city.ac.uk | |
| | Madi | Azhmaganbetov | Madi.Azhmaganbetov@city.ac.uk | |
| | Mohammed | Alom | Mohammed.Alom.2@city.ac.uk | |
| | Rayyan | Zerrouk | Rayyan.Zerrouk@city.ac.uk | |
| | Sarah | Bugshan | Sarah.Bugshan@city.ac.uk | |
| 24 | Atahan | Sahlan | Atahan.Sahlan@city.ac.uk | Warren.Fernando.2@city.ac.uk |
| | Jeremy | Yuen | Jeremy.Yuen@city.ac.uk | |
| | Liana | Chowdhury | Liana.Chowdhury@city.ac.uk | |
| | Mohamed | Kazoz | Mohamed.Kazoz@city.ac.uk | |
| | Mohammed | Koreshi | Mohammed.Koreshi@city.ac.uk | |
| | Vladimir | Socolnic | Vladimir.Socolnic@city.ac.uk | |
| 25 | Adam | Hassan | Adam.Hassan@city.ac.uk | Kaleem.Peeroo@city.ac.uk |
| | Ali | Ay | Ali.Ay@city.ac.uk | |
| | Kamal | Diriye | Kamal.Diriye@city.ac.uk | |
| | Najeeb | Islam | Najeeb.Islam@city.ac.uk | |
| | Usmaan | Mahmood | Usmaan.Mahmood@city.ac.uk | |
| | Valentin | Madzharov | Valentin.Madzharov@city.ac.uk | |
| 26 | Abdul | Jamal | Abdul.Jamal@city.ac.uk | Kaleem.Peeroo@city.ac.uk |
| | Abdulazeez | Jempeji | Abdulazeez.Jempeji@city.ac.uk | |
| | Abdul-Wahid | Mohamed | Abdul-Wahid.Mohamed@city.ac.uk | |

| | | | | |
|----|------------|-----------|------------------------------|------------------------------------|
| | Isa | Munem | Isa.Munem@city.ac.uk | |
| | Kwadwo | Badu | Kwadwo.Badu@city.ac.uk | |
| | Lucy | Nguyen | Lucy.Nguyen@city.ac.uk | |
| 27 | Abishek | Sabesan | Abishek.Sabesan@city.ac.uk | Pedro.Magalhaes-Marques@city.ac.uk |
| | Ammar | Shoaib | Ammar.Shoaib@city.ac.uk | |
| | Edis | Mikulani | Edis.Mikulani@city.ac.uk | |
| | Ismail | Masanja | Ismail.Masanja@city.ac.uk | |
| | Mohamed | El Sheikh | Mohamed.El-Sheikh@city.ac.uk | |
| | Robert | Kutta | Robert.Kutta@city.ac.uk | |
| 28 | Anite | Cangi | Anite.Cangi@city.ac.uk | Pedro.Magalhaes-Marques@city.ac.uk |
| | Dan-Andrei | Coltan | Dan-Andrei.Coltan@city.ac.uk | |
| | Deep | Shah | Deep.Shah@city.ac.uk | |
| | Imtiaz | Ali | Imtiaz.Ali@city.ac.uk | |
| | Mohona | Roy | Mohona.Roy@city.ac.uk | |
| | Tyler | Liddell | Tyler.Liddell@city.ac.uk | |