BSc Business Computing Systems, BSc Computer Science (all strands), MSci (all strands)
Stage 2 – Team Project 2017/18
Module No: IN2018

Developing a Software Product

BAPERS : Bloomsbury's Automated Process Execution Recording System

Student's Brief

Module leader / Lecturer:
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.

Version 1.2
15 Jan. 2018

Contents

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 every stage of the project. The document contains team membership, too. Also, the document includes the case-study that will be used throughout the project.

All students on the BSc Business Computing Systems, BSc Computer Science (all strands), and MSci (all strands) degree programmes undertake a team project in their second year of the UG studies. The only exception to this are the students on the Professional Pathway route. The team project involves analysis, design and implementation 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, the one that cannot be solved by a lone individual in the time available – using the knowledge and skills gained in Stage 1 (e.g. from IN1005 Software Engineering and IN1007 Programming in Java modules, etc.) as well as in Stage 2 (e.g. IN2013 Object-Oriented Analysis and Design, IN2007 Human Computer Interaction (only some students would have attended the latter module), etc.). Yet another purpose, equally important as the ones above, is enhancement of transferable skills including teamwork, communication, negotiation and interpersonal skills.

The module runs in Term 2. Week numbers used in the module are counted from the 1st teaching week of Term 2 and extend for 11 consecutive weeks until the end of the term (including the Reading week in the middle). Some coursework deadlines are scheduled for week 12. This is, in part, so that the students can 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. The following lecture from IN2013 module is, however, useful to consider: "Session 10: Software projects: element of planning, and management of project risk and software change". Also, although it is assumed that teams will deploy and use the UML and other methods formally taught during their courses, but 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 *same project*, but are expected to work independently.

1.2 Learning Objectives

The learning objectives are to demonstrate the ability to plan, organise and conduct a software development project of a 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, the students will gain experience of the following:

- forming a working team, establishing common goals and resolving 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:

- IN1005 – Software Engineering (Stage 1)
- IN1007 – Programming in Java (Stage 1)
- IN1010 – Business Systems (Stage 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
- 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. Hortsmann, *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*, 5th edition, Chapter 23 – Relational Databases, John Wiley and Sons, December 2012.
  - http://www.horstmann.com/bigjava.html
  - There is also a newer, 6th, Edition available.

For furthering your knowledge and skills in Human Computer Interaction, the following is recommended:
- D. Benyon, *Designing Interactive Systems,* Pearsons 2014

Relevant web resources are as follows:
- UML Specification
  - http://www.omg.org/spec/UML/2.3/
  - http://www.omg.org/spec/UML/2.5/
- 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 - http://libguides.city.ac.uk/c.php?g=40522&p=257987.

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 drops out, or lacks commitment, this does not, as a consequence, 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). Assignment to the teams will be done **according to the marks achieved** in Stage 1 (IN1007 mainly, and IN1005), as well as the coursework marks in the IN2013. The students ranked in the top, middle and bottom of the cohort will be as equally represented in each team as possible. Also, **a fair and equal mix between BCS students and ones from a technical, CS (any strand) courses** will be achieved 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 supervisor of the team – ***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 to solve the same problem. 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 (at least in part) – this is up to the teams to decide; Lecture 2 and associated tutorial from IN1005 module (delivered in Stage 1) are especially useful in this respect. Also, the teams **must** use ***UML*** for specification, analysis and design. However, they are free to choose their programming language, database platform, operating system and machine type (provided that *they can supply the right kit, e.g. laptops etc., themselves* in order to demonstrate the results to the TPMT). Traditionally, most teams choose to implement their systems using Java as the programming language, for reasons of availability and familiarity. However, teams are free to use other programming languages, e.g., C++, Python, 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 consultant and the customer (NB this person must contribute to technical work too!);
- ***deputy project manager*** : second in command, and responsible for documentation, reports and standards (NB: this person must contribute to technical work too!);
- ***system analyst*** : responsible for elicitation of customer requirements and requirements analysis and specification;
- ***designer*** : responsible for the system design process and deliverables;
- ***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/system 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 project manager and deputy, of course. An example of typical configuration of roles within a team might be:

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

Despite the roles assigned to each member, it is highly recommended that **all students be actively involved in, and contribute to, the work *throughout the project*.**

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 is referred to as the ***Project Binder*** or ***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 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 form a part of 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 supervisors on demand*. In any case, intermediate versions must be submitted to the consultants at the end of each milestone!* The supervisor 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, Bloomsbury's Image Processing Laboratory (BIPL). The students will have to discover the full requirements by interviewing them. Please see the schedule on Moodle. 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 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 supervisor, i.e. consultant, is to monitor progress of the team, and help resolve difficulties and risks communicated to him or her. Team supervisors *are not meant* to act as technical consultants to the teams. This is so that fairness is provided to all teams. Please note that not all consultants are versed in every part of the Team Project tasks. If any such 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. In any case, the consultant is *on the side of the team*, and will assist in resisting unreasonable demands from the Customer.

2.4 Communication

Most communication during the Team Project will be done by electronic 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 have exclusive 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/supervisor – the latter type of meetings – *weekly* in person. Please note that team consultants are very busy (they are mostly PhD students), and will need sufficient notice of appointments. A consultant may only be able to grant appointments at certain times/days, but will see each team he/she supervises for ***up to*** **30 minutes every week**. For consultant meetings, the most effective way is to have a dedicated date 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 (29 January – 13 April), including the Reading week, and some coursework will be due in week 12 to give you sufficient time to succeed. The deadlines have been communicated to the lecturers of all other Stage 2, Term 2 modules also, in order to give you appropriate time to succeed in all modules. The timetable of activities and deliverables (outputs) is given in Table 1.

Week	Briefings/Q&As/Deadlines	Activities	Deliverables
1	**30 January:** Briefing on module organisation **1 February:** Briefing on Requirements Specification. **NB: There are two 2-hour lectures (Briefings) in Week 1. This is for students' benefit – to help them succeed in the project by offering an additional Briefing early in the module.**	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	None
1, 2	**06 – 07 February:** Interviews **08 February:** Briefing on System Design	Interviews with customer - further requirements elicitation. Students **must** attend the respective interview. See the schedule on Moodle	None
1–5	**15 February:** Version Control, Database modelling recap and Q&A session **22 February:** Java GUI recap and Q&A **01 March:** Briefing on System Implementation **Sun. 04 March, 5pm** (end of week 5)**:** *Submission* of: "Requirements Specification and System Design" document, and Project Binder and Individual Diaries.	Specifying system requirements and system design.	"Requirements Specification and System Design" document – formal, summative assessment to be submitted on Moodle. Individual Diaries and Team Project Binder (to your consultant)
6–12	**15 March:** Guest talk on Agile Methodology and Test-Driven development **22 March:** Briefing on Database Connectivity programming (JDBC), and General Feedback on System Design **5 April:** Guest talk on how to make an effective presentation **TBC: Week ending 15 April:** *Demonstration* of the software in operation – summative assessment, and the Showcase event (Thur. 12th Apr, Whole day); **TBC: Week ending 22 April:** *Submission* of Implementation Reports (Mon. 16 April, 5pm); *Team presentation* (Tue. 17th Apr); *Submission* of Individual Reports; (Sun. 22 April, 5pm); **TBC: Mon. 23 April, 5pm** Final versions of Project Binder and Individual Diaries, and final evidence about contribution by any of the team members!	System Implementation and Testing. Attendance to the Demo and Showcase event Writing Implementation and Individual Reports Preparation and delivery of the team project presentation.	Formal, summative assessments: 1. Working system demonstration ("Demo") 2. Implementation Report 3. Team presentation 4. Individual Report Individual diaries and Team Project Binder (to your consultant)

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

Formal Briefing materials (slides) will be prepared and presented in class during the team project. 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 – 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 (30 January) – Introduce Team Project and its structure, e.g. team dynamics, etc.
- Week 1 (01 February) – What is a requirements specification?
- Week 2 (08 February) – What is a system design?
- Week 3 (15 February) – Version control, Database modelling, and Q&A
- Week 4 (22 February) – Java GUI recap, and Q&A
- Week 5 (01 March) – What is software implementation?
- Week 7 (15 March) – Guest talk on Agile Methodology and Test-Driven development
- Week 8 (22 March) – Java JDBC programming
- Week 9 (29 March) – Preparation for Software Demo and Team presentations
- Week 10 (05 April) – Guest talk on how to make an effective (team) presentation
- Week 11 (12 April) – Software Demo and Showcase event

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 a 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 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 the approach 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.

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 and deployed,
   - 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, which mainly consists of checking if the functional requirements are met. Usability of the software will be evaluated 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 a table of all coursework components to be delivered by each team (and individual), together with the maximum mark that can be awarded:

Item	Deliverables	Week	Mark (%)
1	Requirements Specification and System Design document	5	40
2	System Demonstration, Implementation Report and Team Project Presentation	11, 12	50
3	Individual Report	12	10
	Total		100

Table 2: Team Project Deliverables and the 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 be, however, **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 milestone, e.g. in the class (general feedback), during Q&A sessions, during lecturer's Office Hours 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 particular circumstances, the consultant might decide to involve the lecturer in some cases.

Additionally, during the course of the project and based on clear evidence from a variety of sources (see below), the consultant might 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 he/she can form a full and correct view.

VERY IMPORTANT
At the ***end*** of the team project the ***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, 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 must 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 supervisor;
- notes from the team supervision 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 deadline.
This procedure guarantees **a fair marking approach for each team member**!

5. Submission of Deliverables

For summative assessment, there are two deliverables/documents[1] to be delivered by each team (Requirements Specification and System Design deliverable, and Implementation Report deliverable), and one to be delivered personally by every student (Individual Report).

[1] The teams will also need to prepare two pieces of summative assessment that are not documents: Demo of the working product and Team presentation.

All summative assessment documents must be submitted ***electronically via Moodle*** **before the respective deadline**.
Where more than one deliverable is due on a given date they should be submitted as separate documents. The dates for submitting deliverables are listed in Table 1 and 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 department's and the university's regulations for project, coursework-only modules, see http://www.city.ac.uk/__data/assets/pdf_file/0008/68921/assessment_and_feedback_policy.pdf .
In addition, after each of the two phases – Requirements & Design and Implementation – each team must submit the Project Binder, and each student must submit the respective Individual Diary to the 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 contribution (see e.g. Section 4).

6. Assessment Criteria for Deliverables

6.1 General criteria for documents

Written documents submitted for marking will be assessed using the general criteria covering presentation and style. In addition to this, each particular 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 person(s).
10% of the marks awarded for Requirements Specification & System Design document, and Individual Reports will be awarded for General Criteria.

General Criteria:

- Appropriate title, page numbering and version control (2)
- Introduction, and Purpose & Scope of the document (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
    - Description of the existing system (what the company have). This should simply be in natural language, using your own words. Students should not "design" the current system; (5)
    - Full use case diagram(s) defining collections of use cases and their interactions with actors covering the whole functionality of the system; (15)
    - 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. **Important**: the teams are NOT allowed to provide UC specifications for any of the UCs from Tutorial 1 of OOAD (IN2013) module; (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, exceptions, set and get operations, constructors and destructors. 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 with meaningful values, and all necessary SQL DML statements which are needed to create required reports – **2** non-trivial reports pertaining to the case-study – must be provided.
  - The class diagram is not the same as database schema. The teams are expected to use Relational Database Management Server (RDBMS), not an OO database server. 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 menus/forms etc. Some relevant material for this section has been covered, for example, in IN2007 – Human Computer Interaction (HCI), module, which is core for BCS programme students. (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 non-expert terms (i.e., use cases) complemented by technical specifications represented in a structured fashion (class diagram).

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

Accompanying the final working 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, in electronic form should accompany the Implementation Report (i.e. in a single zip file on Moodle). ***Failure to supply the source code will result in zero marks being awarded for the whole of the Working Product stage.***

A presentation about the project should be prepared and delivered: (10)

- Content (up to 5 marks)
- Presentation skills/style (up to 5 marks)

The working product will be assessed using the following criteria:

- Working product Demo
  - Completeness and quality of the product in relation to the requirements (70)
  - Ease of use and consistency of the GUI (5)
- Implementation Report
  - Software architecture/Compilation/Run-time components (6)
  - Testing plans and reports (9)

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

- 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 staff to award marks for the presence of features in the product which are determined 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 demonstration will ask the demonstrator/team to show features in accordance with the requirements.

The software demonstration ("Demo"), and the Showcase event, is planned to take place in the Tait Building Basement (lower ground floor) on the 12th April. You will need to bring ***your own machine*** to demo the software you implemented. The details about the Demo and the Showcase event (including the schedule, etc.) will be ***confirmed at least 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. 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 particular problems personally encountered and how they could be avoided in future. (15)
- Description of how the team worked together, providing evidence-based assessment about the contribution to the team deliverables of the other members. (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 detected[2], all students/teams involved will be given **zero** marks for the copied deliverables.

8. BAPERS Requirements

The owner of the 'The Lab', Bloomsbury's Image Processing Laboratory (**BIPL**), Mr Glynne Lancaster, sends out an invitation to tender for the development of some software for use in the laboratory. Being interested in tendering for the contract, your company has made enquiries and has received the following outline of **BIPL**'s requirements for Bloomsbury's Automated Process Execution Recording System (BAPERS).

Bloomsbury's Automated Process Execution Recording System (BAPERS)

BIPL is a photographic laboratory, which handles the work of professional photographers, and must work to tight deadlines without sacrificing quality. It offers to its customers a wide variety of standard *jobs*, each of which requires one or more standard *tasks* to be performed by its expert laboratory staff. In addition, **BIPL** will always respond to the best of its ability to special instructions from its customers regarding particular *jobs*.

At present, **BIPL** technicians can perform around 30 *standard tasks*, each of which has an identifier, and is carried out at a particular processing station or location within the laboratory. For example, see the table below:

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

Task ID	Task Description	Location	Shelf slot	Price (£)	Duration (min)
1.	Use of large copy camera	Copy Room	CR25	19.00	120
2.	Black and white film processing	Development area	DR12	49.50	60
3.	Bag up	Packing Departments	PR10	6.00	30
4.	Colour film processing	Development Area	DR25	80.00	90
5.	Colour Transparency processing	Development Area	DR100	110.30	180
6.	Use of small copy camera	Copy Room	CR16	8.30	75
7.	Mount Transparencies	Finishing Room	FR5	55.50	45
…	Etc.	Etc.	Etc.	Etc.	Etc.

Table 3. An excerpt from the set of tasks handled by BIPL staff

Once the task is completed, the semi- or fully-completed job is placed in a shelf slot from where it can be collected for further processing or dispatched to the customer.

As image technology improves, it is expected that new processes will become available, requiring the installation of new equipment and an increase in the number of standard tasks. Thus, BAPERS **must** offer functionality for adding new tasks, removing or updating existing tasks.

BIPL carries out *jobs* on behalf of its customers. Each job is given a unique identifier on receipt. In addition, special instructions can be required for a specific job. Urgent jobs must be completed within 6 hours, while the normal jobs within 24 hours. But, the customer may stipulate a deadline for completion of 3 hours (at 100% surcharge) or shorter (at an even higher rate).

BIPL operates on the basis of high-turnover, low profit margin. Most of the jobs that it handles are priced below £400. At any given time, hundreds of jobs will be in progress or pending within the laboratory.

An important point is that every job accepted must be chargeable to a valid customer account, either an existing customer account, or a newly created customer account (e.g. a photographer may walk in, leave film to be developed and pay cash on collecting the finished job).

BIPL want to enable the employee on the reception desk to enter the job on a computer terminal. The material will be labelled with the job number and taken down to the laboratory.

The laboratory staff will interrogate **BAPERS** to ascertain the tasks required. As the job is transferred from one location to another in the laboratory, the staff responsible for each task will record its completion on a computer terminal in their location before passing it on. A terminal will be required in each of the Copy Room, Dark Room, Development Area, Printing Room, Finishing Room, and Packing Department.

Many jobs will be going through the laboratory at any given time, and confusion between them must be avoided. At all cost, loss or mistreatment of the customer's material must not occur. Queues of work may build up at the processing stations, but flexible scheduling is required to allow priority to be given to *urgent* jobs (important customer, tight deadline!), over the *regular/normal* jobs. The system should provide functionality for inspecting the list of active/pending jobs as well as already completed ones, including the inspection of the progress of individual tasks (active/pending tasks vs. completed ones).

BAPERS must therefore provide the following main facilities:

BAP-ACCT *Accept job at reception*: Identify existing customer account, or create new customer account. Assign job number. Print label and receipt for materials submitted. Record deadline for completion and any special instructions. Alert laboratory Shift Manager to arrival of a new job. This functionality will be mainly used by the receptionists but can also be made accessible by Office Manager or Shift Manager in case receptionist(s) is/are absent.

BAP-PROC *Process job through laboratory*: Respond to enquiries from any computer terminal about status of a job, or of all jobs that are "in progress", or of all jobs (including the completed ones). Update status of any given job by recording completion of current task and commencement of next (possibly with transfer of material to a new location). This functionality is available to Technicians, Shift and Office Manager.

In addition, alert Shift and/or Office Manager (by, for example, displaying a visual cue with appropriate text) if the expected time to complete outstanding tasks for any job is likely to exceed the set time period, i.e. if the deadline for the job is not likely to be met; the alerts should be performed only for these two user roles.

BAP-REPT *Producing various reports*. The following reports are required: i) Individual report for the jobs brought in by a particular customer for an arbitrary period of time as specified at the time of generating the report (e.g. per month); ii) individual performance report on work undertaken by a member of BIPL staff, and iii) summary performance report for work undertaken by BIPL during day and night shifts.

BAP-PAYM *Payment processing and dealing with debtors*. The customers are supposed to pay once the jobs they had placed have been completed. Customers can pay by cash or credit/debit card only. **Important**: The system is **not** connected to an external payment processor. However, payment records are stored by the system; at least the payment amount and type of payment need to be stored. If a card payment is successful, the following details are recorded too: expiry date, type, and the last 4 digits of the card used. Cash payments are recorded, too.

Valued customers are allowed to pay by the 10th of each month for their jobs completed by the end of the previous calendar month. They can make a single payment for several jobs, e.g. all jobs accumulated by the end of the month, or make a single payment for a particular job (i.e. each job is paid separately, but in full).

The late payments should be detected automatically by the system and Office Manager should be alerted. If the Office Manager is logged-in, the system will generate alerts as pop-up windows, warning at regular intervals of 15 minutes until the Office Manager acknowledges the receipt of the warning.

If a valued customer fails to clear the outstanding balance by the 20 of the next month, they will be sent a Reminder letter (the First reminder letter). The system should generate the late payment reminder letters for each customer automatically. The reminder letters are stored until the Office Manager logs in. Once the Office Manager logs in the system will offer the Office Manager the opportunity to print the letters. If a month after the first reminder letter had been sent the outstanding balance has not been cleared, the system should generate a Second reminder letter for the customer automatically and *suspend their account*, i.e. the customer will not be able to get any further jobs processed. If a month after the valued customer's account has been suspended the outstanding balance still has not been cleared, the system should mark the valued customer account as 'in default'. BAPERS should generate on demand a list of all customers whose accounts are 'in default' for taking a legal action against them. Receiving a payment from a customer with a suspended account will reactivate the account automatically. A customer account marked as 'in default' can only be reactivated by the Office Manager. Office Manager should be alerted for these subsequent late payment reminders in the same way as for the first reminders (i.e. pop-up windows). The same printing mechanisms and options should also be implemented as for first reminders. In addition to automatic generation, the system should offer ways for the reminders to be printed on demand either as a batch for all reminders, a specific type (e.g. all first reminders) or individual reminders.

BAP-CUST *Valued customers* may be given a discount. The specific terms for giving a discount are defined in a *discount plan* set up for a valued customer (i.e. a customer who uses the services provided by **BIPL** frequently, and pays on time). Customer accounts are upgraded to 'valued customer' status at the discretion of the Office Manager (and can be downgraded too). In addition, the Office Manager decides what particular discount plan a valued customer should have**.** A valued customer can be associated with only a single discount plan at any given time. **BIPL** see the need for three different types of discount plans to be made available in **BAPERS** for valued customers:

i) **Fixed discount** – the same percentage of discount is given to the valued customer for each job. The value of this discount is calculated and deducted from the value of the job at the time of accepting the job;

ii) **Variable discount** – the percentage of the discount is set for each *task* and may vary between the tasks. The value of this discount is calculated and deducted from the value of the job at the time of accepting the job.

iii) **Flexible discount** – the percentage of the discount depends on the values of the jobs by the same customer accumulated within a calendar month. In general, a single flexible discount plan can have a number of "bands", and respective discount rates, associated with it. The overall applicable discount should be possible to calculate by BAPERS both: i) at the end of the month, or ii) on demand. The value of the discount is paid back to the customer by deducting the corresponding amount from the value of future jobs.

BAP-ADMN *Administering the system.* This includes creating a user account for BAPERS and setting up access privileges. Also operating database backups and restores need to be implemented – these must be available both on demand and automatically. The following user roles are essential to be implemented: Office Manager, Shift Manager, Receptionist and Technician.

Also, it is important that an appropriate Concurrency Control mechanism is used in the system so that database inconsistencies are avoided. For this, an appropriate isolation level between concurrent database transactions is needed. For example, since several employees may be working at different terminals, trying to access the same tasks and (possibly) working on the same jobs, it is an essential requirement to have proper concurrency control in place to resolve conflicting requests. All BAPERS activities and calculations must leave the database in a consistent state.

BAPERS is required as soon as possible in order for **BIPL** to continue to offer its outstanding service to its growing customer base. **BIPL**'s mission statement is **"Perfect results, on time, every time"**.
Your consultant has won the contract from **BIPL** to assess their requirements and develop *a prototype of BAPERS which runs on a single laptop/computer but provides all functionality required* from **BAPERS***.* Your team has been tasked with doing the job.
To fully understand all BAPERS requirements you **must** interview Mr Lancaster, the owner of the Lab. Before the interview, you must be clear with the content of this document and other material/information communicated to you regarding the BAPERS requirements, if any. The interview dates and times have been arranged already.

8.1 Relationship between the case study in OOAD (IN2013) and Team Project (IN2018)

The BAPERS case-study in Team Project is closely *related* to the version of the case study used in OOAD (IN2013) module. This was purposefully done. You ought to appreciate this extra link between IN1005 (Software Engineering), IN2013 (Object-oriented Analysis and Design) and IN2018 (Team Project) modules: the modules are connected via a similar case-study used, not only via the related material taught. Also, this gives you an opportunity to appreciate further the need for a structured approach to software development for a relatively complex piece of software, e.g. slight additions to, or changes of, the initial requirements as presented in OOAD module.

Please note that the scope and functionality of BAPERS has been **extended**, and in some parts **modified**, for the purposes of Team Project. Thus, you must make sure you follow the description in *this* document, and all the information obtained during the communication in Team Project (e.g. interviews, Q&A session, etc.), when developing BAPERS in Team Project module. Therefore, you need to read the description of BAPERS in this document very carefully. You should, of course, use the knowledge and skills obtained in OOAD module, including the tutorial answers etc., but please note that the assessment tasks in Team Project are not the same as the ones in OOAD module. For example, class diagram is required in both, but in Team Project you are asked to i) develop the software based on a version of BAPERS available in Team Project, ii) you are asked to develop a complete Design stage class diagram (despite parts of it being covered in OOAD tutorials – these were for a different version of BAPERS), etc.

9. Documents provided by customer

9.1. By Mr Glynne Lancaster, Owner

JOB SHEET 1245 **Date 12/01/2018**

Customer: ACC3005
Description of work:
Estimated time for collection: 4 pm, 13 January 2018

Description of work in progress:

Code	Price, £	Task	Department	Start Time	Time Taken	Completed by	Shelf on completion
ABN54	19.00	1	Copy Room	12:00	20 min	John Nash	CR205
		2	Development	12:30	40 min	Lee Hong	DR42
		3	Packing	13:20	10 min	Marina Scott	PK102
ACN54	19.00	1	Copy Room	12:20	35 min	John Nash	CR143
		4	Development	13:10	40 min	Lee Hong	DR31
		3	Packing	14:00	10 min	Julie Abbot	PK100
ACT108	96.00	1	Copy Room	12:55	1 h 40 min	John Nash	CR120
		5	Development	14:40	30 min	Lee Hong	DR32
		3	Packing	15:20	10 min	Julie Abbot	PK101
ACT35	20.00	6	Copy Room	14:40	20 min	John Nash	CR210
		5	Development	15:10	20 min	Lee Hong	DR30
		7	Finishing Room	15:35	25 min	Julie Abbot	FR11
B108	8.30	2	Development	12:20	30 min	Stewart Pask	DR34
		3	Packing	12:50	10 min	Julie Abbot	PK108
C108	8.30	4	Development	13:10	20 min	Stewart Pask	DR35
		3	Packing	13:35	10 min	Julie Abbot	PK106

Account Details

Account Holder Name: City University
Account No: ACC0001
Contact Name: Prof David Rhind
Address: Northampton Square, London EC1V 0HB
Phone: 0207 040 8000
Agreed Discount: Fixed
Discount Rate: 3%

Account Holder Name: AirVia Ltd
Account No: ACC0002
Contact Name: Mr Boris Berezovsky
Address: 12, Bond Street, London WC1V 8HU
Phone: 0207 321 8523
Agreed Discount: Flexible (per task)
Discount Rate:

Task	Discount Rate
1. Use of large copy camera	1%
1. Black and white film processing	1%
2. Bag up	0 %
3. Colour film processing	2 %
4. Colour Transparency processing	2 %
5. Use of small copy camera	0 %
6. Mount Transparencies	2%

Account Holder Name: InfoPharma Ltd
Account No: ACC0003
Contact Name: Mr Alex Wright
Address: 25, Bond Street, London WC1V 8LS
Phone: 0207 321 8001
Agreed Discount: Variable (on volume per month)
Discount Rate:

Volume	Discount Rate
< £1000	: 0 %
£1000 - £2000	: 1%
£2000+	: 2 %

The Lab
2, Wynyatt Street
London, EC1V 7HU
Phone: 0207 235 7534

Invoice 30123 / 13/01/2018

Account: ACC0001

Account Holder Name:	City University
Account No:	ACC0001
Contact Name:	Prof David Rhind
Address:	Northampton Square, London EC1V 0HB
Phone:	0207 040 8000

Job No: 1245 Completed: 13 January 2018

Description of work:

Code	Job Description	Price (£)	Task IDs
ABN54	5 x 4 B& W copy negatives	19.00	1, 2, 3
ACN54	5 x 4 Colour copy negatives	19.00	1, 4, 3
ACT108	10 x 8 Colour copy transparency	96.00	1, 5, 3
ACT35	35 mm Colour copy transparency	20.00	6, 5, 7
B108	10 x 8 processing	8.30	2, 3
C108	10 x 8 C41 processing	8.30	4, 3
Sub-Total		170.60	
Discount agreed:		3%	
		165.48	
Total payable (VAT at 20%)		198.58	

Make a payment within 30 days by cash, card or bank transfer to our account:

‘The Lab’, Bloomsbury’s Image Processing Laboratory
Barclays Plc, City University Branch, 10, Northampton Square
Sort Code 30-20-70,
Account number: 67103456

9.2. By Mr Rick Evans, Office Manager

Individual Performance Report

Period: 13/01/2018 – 13/01/2018

Name	Code	Task IDs	Department	Date	Start time	Time taken	Total
John Nash	ABN54	1	Copy Room	13/01/2018	12:00	20 min	2h 55 min
	ACN54	1	Copy Room	13/01/2018	12:20	35 min	
	ACT108	1	Copy Room	13/01/2018	12:55	1 h 40 min	
	ACT35	1	Copy Room	13/01/2018	14:40	20 min	
Lee Hong	ABN54	2	Development	13/01/2018	12:30	40 min	2 h 10 min
	ACN54	4	Development	13/01/2018	13:10	40 min	
	ACT108	5	Development	13/01/2018	14:40	30 min	
	ACT35	5	Development	13/01/2018	15:10	20 min	
Julie Abbot	ACN54	3	Packing	13/01/2018	14:00	10 min	1 h 05 min
	ACT108	3	Packing	13/01/2018	15:20	10 min	
	ACT35	7	Finishing Room	13/01/2018	15:35	25 min	
	B108	3	Packing	13/01/2018	12:50	10 min	
	C108	3	Packing	13/01/2018	13:35	10 min	
Marina Scott	ABN54	3	Packing	13/01/2018	13:20	10 min	0 h 10 min
Stewart Pask	B108	2	Development	13/01/2018	12:20	30 min	0 h 50 min
	C108	4	Development	13/01/2018	13:10	20 min	
Total: effort:							7 h 10 min

Summary Performance Report

Period: 13/01/2018 – 20/01/2018

Day shift 1 (5:00 am – 2:30 pm)

Date	Copy Room	Development	Finishing	Packing
13/01/2018	8 h 20 min	10 h 45 min	8 h 30 min	2 h 30 min
14/01/2018	10 h 20 min	11 h 30 min	10 h 40 min	1 h 30 min
15/01/2018	7 h 20 min	13 h 20 min	9 h 30 min	2 h 50 min
16/01/2018	5 h 10 min	15 h 20 min	6 h 20 min	2 h 20 min
17/01/2018	7 h 10 min	10 h 50 min	7 h 20 min	3 h 25 min
18/01/2018	9 h 25 min	14 h 30 min	9 h 20 min	2 h 50 min
19/01/2018	10 h 30 min	12 h 40 min	11 h 30 min	1 h 30 min
20/01/2018	7 h 30 min	11 h 10 min	8 h 20 min	2 h 50 min
Total	65 h 45 min	100 h 05 min	71 h 30 min	19 h 45 min

Day shift 2 (2:30 pm – 10 pm)

Date	Copy Room	Development	Finishing	Packing
13/01/2018	8 h 20 min	13 h 45 min	12 h 30 min	2 h 30 min
14/01/2018	12 h 20 min	15 h 30 min	6 h 40 min	1 h 30 min
15/01/2018	10 h 20 min	16 h 20 min	7 h 30 min	3 h 20 min
16/01/2018	6 h 10 min	17 h 10 min	5 h 20 min	2 h 20 min
17/01/2018	4 h 10 min	11 h 50 min	8 h 20 min	4 h 25 min
18/01/2018	13 h 25 min	10 h 30 min	9 h 20 min	2 h 50 min
19/01/2018	12 h 30 min	16 h 20 min	10 h 30 min	3 h 30 min
20/01/2018	10 h 30 min	13 h 10 min	10 h 20 min	1 h 50 min
Total	77 h 25 min	114 h 35 min	70 h 30 min	22 h 15 min

Night shift 1 (10 pm – 5 am)

Date	Copy Room	Development	Finishing	Packing
13/01/2018	8 h 20 min	5 h 45 min	8 h 30 min	1 h 30 min
14/01/2018	6 h 20 min	7 h 30 min	6 h 40 min	0 h 30 min
15/01/2018	3 h 20 min	2 h 20 min	5 h 30 min	1 h 50 min
16/01/2018	2 h 10 min	4 h 20 min	3 h 20 min	1 h 20 min
17/01/2018	1 h 10 min	3 h 50 min	3 h 20 min	1 h 40 min
18/01/2018	7 h 25 min	5 h 30 min	4 h 20 min	1 h 30 min
19/01/2018	7 h 30 min	4 h 40 min	6 h 30 min	1 h 30 min
20/01/2018	3 h 30 min	7 h 10 min	5 h 20 min	2 h 00 min
Total	39 h 45 min	41 h 05 min	43 h 30 min	11 h 50 min

For period (13/01/2018 – 20/01/2018)

Date	Copy Room	Development	Finishing	Packing
Day Shift 1	65 h 45 min	100 h 05 min	71 h 30 min	19 h 45 min
Day Shift 2	77 h 25 min	114 h 35 min	70 h 30 min	22 h 15 min
Day Shift 3	39 h 45 min	41 h 05 min	43 h 30 min	11 h 50 min
Total	182 h 55 min	255 h 45 min	185 h 30 min	53 h 50 min

Level of access to BAPERS

The access to the system is limited to the laboratory staff only; customers cannot access the system.

Office Manager: Access to all subsystems of BAPERS.
Shift Manager: Access to all subsystems of BAPERS except BAP-PAYM, BAP-CUST and BAP-ADMN.
Receptionist: Access to BAP-ACCT
Technician: Access to BAP-PROC

Invoice Reminders

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

The Lab

Bloomsbury's Image Processing Laboratory
2, Wynyatt Street, London, EC1V 7HU
Phone: 0207 235 7534

18th January 2018

Dear Mr. Smith,

REMINDER - INVOICE NO.: 197362
Job No: 2345 Total Amount: 294.46

According to our records, it appears that we have not yet received payment of the above invoice, which was posted to you on 18th December 2017, for photographic work done in our laboratory.

We would appreciate payment at your earliest convenience.

If you have already sent a payment to us recently, please accept our apologies.

Yours sincerely,

G. Lancaster

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

The Lab

Bloomsbury's Image Processing Laboratory
2, Wynyatt Street, London, EC1V 7HU
Phone: 0207 235 7534

18th February 2018
Dear Mr. Smith,

FINAL REMINDER - INVOICE NO.: 197362

Job No: 2345 Total Amount: 294.46

It appears that we still have not received payment of the above invoice, which was posted to you on 18th December 2017, for photographic work done in our laboratory, despite a reminder letter posted to you 1 month later.

Unless you pay the outstanding amount in full within SEVEN DAYS, or contact us with proposals for repayment, we will have no option but to refer the matter to our solicitor.

Please send payment immediately to avoid further action.

Yours sincerely,

G. Lancaster

10. Team membership

Team	First Name	Surname	Email ID	Course	Consultant
1	Dominic	Findlay	Dominic.Findlay@city.ac.uk	USCSCI	Remilekun.Basaru.1@city.ac.uk
	Jean-Louis	Gosselin	Jean-Louis.Gosselin@city.ac.uk	USGTEC	
	Zerbu	Bulut	Zerbu.Bulut@city.ac.uk	USCSCI	
	Richard	Ho	Richard.Ho@city.ac.uk	USCSCI	
	Kajol	Pandya	Kajol.Pandya@city.ac.uk	USCOSC	
	Sophie	Scribbins	Sophie.Scribbins@city.ac.uk	USBCSW	
2	Gaznish	Jassal	Gaznish.Jassal@city.ac.uk	USCSCI	Remilekun.Basaru.1@city.ac.uk
	Abdurahman	Omar	Abdurahman.Omar@city.ac.uk	USCSCI	
	Prem	Grewal	Prem.Grewal@city.ac.uk	USCSCI	
	Chandni	Bhojani	Chandni.Bhojani@city.ac.uk	USBCSW	
	Usama	Mahmood	Usama.Mahmood@city.ac.uk	USCSCI	
	Saiyuaran	Vipulananthan	Saiyuaran.Vipulananthan@city.ac.uk	USCSCI	
3	Serkan	Sahin	Serkan.Sahin@city.ac.uk	USCSCS	Remilekun.Basaru.1@city.ac.uk
	Haleema	Kanwal	Haleema.Kanwal@city.ac.uk	USBCSW	
	Matthew	Russell	Matthew.Russell@city.ac.uk	USCSCI	
	Nayim	Nurdin	Nayim.Nurdin@city.ac.uk	USCSCI	
	Shawn	Collingwood	Shawn.Collingwood@city.ac.uk	USCSCI	
	Andrei	Solot	Andrei.Solot@city.ac.uk	USCSGT	
4	Joshua	George	Joshua.George@city.ac.uk	USGTEC	Remilekun.Basaru.1@city.ac.uk
	Tilak	Pindoria	Tilak.Pindoria@city.ac.uk	USBCSW	
	Natalia	Kujawka	Natalia.Kujawka@city.ac.uk	USCOSC	
	Luv	Seewoosurran	Luv.Seewoosurran@city.ac.uk	USCSCI	
	Devante	Kumar	Devante.Kumar@city.ac.uk	USCSCI	
	Armani	Bashorun	Armani.Bashorun@city.ac.uk	USCSCI	
5	James	Kayes	James.Kayes.2@city.ac.uk	USCSCI	Remilekun.Basaru.1@city.ac.uk
	Alberto	Brusa	Alberto.Brusa@city.ac.uk	USCSGT	
	Muhammad	Patel	Muhammad.Patel.2@city.ac.uk	USCSCI	
	Adedamilola	Adesanya	Adedamilola.Adesanya@city.ac.uk	USGTEC	
	Shaumik	Rahman	Shaumik.Rahman@city.ac.uk	USCSCI	
	Faisal	Alibrahim	Faisal.Alibrahim@city.ac.uk	USBCSW	
6	Christopher	Michael-Stokes	Christopher.Michael-Stokes@city.ac.uk	USCSCI	Konstantin.Pozdniakov.1@city.ac.uk
	Kwok	Law	Kwok.Law@city.ac.uk	USCSCI	
	Mugheeth	Razzaque	Mugheeth.Razzaque@city.ac.uk	USCSCI	
	Patryk	Gulbinovic	Patryk.Gulbinovic@city.ac.uk	USCSCI	
	Gerda	Morockaite	Gerda.Morockaite@city.ac.uk	USCSCI	
	Mihai-Gabriel	Baeasu	Mihai-Gabriel.Baeasu@city.ac.uk	USBCSW	
7	Felice	Gregorio	Felice.Gregorio@city.ac.uk	USCSCI	Konstantin.Pozdniakov.1@city.ac.uk
	Adem	Aybar	Adem.Aybar@city.ac.uk	USGTEC	
	Usaamah	Mahmood	Usaamah.Mahmood@city.ac.uk	USCSCI	
	Kamran	Saeed	Kamran.Saeed@city.ac.uk	USBCSW	
	Kimberly	Eslami	Kimberly.Eslami@city.ac.uk	USCSCI	
	Allen	Kurshumlija	Allen.Kurshumlija@city.ac.uk	USCSCI	
8	Elmar	Talibzade	Elmar.Talibzade@city.ac.uk	USGTEC	Konstantin.Pozdniakov.1@city.ac.uk
	Mihai-Alexandru	Matraguna	Mihai-Alexandru.Matraguna@city.ac.uk	USCOSC	
	Cosmin	Capatina	Cosmin.Capatina@city.ac.uk	USGTEC	
	Elias	Thiebaut	Elias.Thiebaut@city.ac.uk	USBCSY	
	Uzma	Shad	Uzma.Shad@city.ac.uk	USCSCS	
	Kimberley	Nguyen	Kimberley.Nguyen@city.ac.uk	USGTEC	

9	Connor	Gowdy	Connor.Gowdy@city.ac.uk	USCSGT	Konstantin.Pozdniakov.1@city.ac.uk
	Redi	Ymeri	Redi.Ymeri@city.ac.uk	USCSCI	
	Mariwan	Ahmed	Mariwan.Ahmed@city.ac.uk	USCSCI	
	Gursavek	Singh	Gursavek.Singh@city.ac.uk	USBCSW	
	Zakaria	Sheikh	Zakaria.Sheikh@city.ac.uk	USCSCI	
	Michael	Da Silva Macedo	Michael.Da-Silva-Macedo@city.ac.uk	USCSCI	
10	Kavishan	Prabarajah	Kavishan.Prabarajah@city.ac.uk	USCSCI	Swetha.Bobba.1@city.ac.uk
	Archie	Bradshaw	Archie.Bradshaw@city.ac.uk	USGTEC	
	Abarna	Rubakrishnan	Abarna.Rubakrishnan@city.ac.uk	USBCSW	
	Nicholas	Casagrande	Nicholas.Casagrande@city.ac.uk	USCSCI	
	Nimshi	Fernando	Nimshi.Fernando@city.ac.uk	USCSCI	
	Shahbaz	Raja	Shahbaz.Raja@city.ac.uk	USCSCS	
11	Cameron	Ellis	Cameron.Ellis@city.ac.uk	USCSGT	Swetha.Bobba.1@city.ac.uk
	Joseph	Rullan	Joseph.Rullan@city.ac.uk	USGTEC	
	Ho	Wong	Ho.Wong.2@city.ac.uk	USCSCI	
	Ivo	Kulesza	Ivo.Kulesza@city.ac.uk	USBCSW	
	Lukas	Ziura	Lukas.Ziura@city.ac.uk	USCSCI	
	Adam	Patel	Adam.Patel@city.ac.uk	USCSCS	
12	Cristian	Ghita	Cristian.Ghita.2@city.ac.uk	USGTEC	Swetha.Bobba.1@city.ac.uk
	Kane	Dharan	Kane.Dharan@city.ac.uk	USBCSW	
	Ahmed	Abukar	Ahmed.Abukar@city.ac.uk	USCSCI	
	Karan	Dixit	Karan.Dixit@city.ac.uk	USCSCI	
	David	Gomes Pinto Madaleno	David.Gomes-Pinto-Madaleno@city.ac.uk	USCSCI	
	Aimal	Hederzada	Aimal.Hederzada@city.ac.uk	USGTEC	
13	Oliver	Brown	Oliver.Brown@city.ac.uk	USCSCI	Marius.Braun.1@city.ac.uk
	Jean	Pascal	Jean.Pascal@city.ac.uk	USCSCI	
	John	Abedejos	John.Abedejos@city.ac.uk	USCSCI	
	Maria	Vanegas Silva	Maria.Vanegas-Silva@city.ac.uk	USCSCI	
	Fabio	Fernandes	Fabio.Fernandes@city.ac.uk	USCSCI	
	Gianfranco	Chan	Gianfranco.Chan@city.ac.uk	USBCSW	
14	Junad	Ahmed	Junad.Ahmed@city.ac.uk	USCSCI	Marius.Braun.1@city.ac.uk
	Gagandip	Flora	Gagandip.Flora@city.ac.uk	USCSCI	
	Hasan	Al-Allaq	Hasan.Al-Allaq@city.ac.uk	USCSCI	
	Jan	Andrzejczyk	Jan.Andrzejczyk@city.ac.uk	USCSCI	
	Aulon	Vishesella	Aulon.Vishesella@city.ac.uk	USCSCI	
	Rameez	Khanzada	Rameez.Khanzada@city.ac.uk	USBCSW	
15	Sai	Uppinakuduru	Sai.Uppinakuduru@city.ac.uk	USCSCS	Marius.Braun.1@city.ac.uk
	Sameer	Ali	Sameer.Ali@city.ac.uk	USGTEC	
	Khesrou	Zemaray	Khesrou.Zemaray@city.ac.uk	USCSCI	
	Warsame	Ali	Warsame.Ali@city.ac.uk	USCSCI	
	Henal	Desai	Henal.Desai@city.ac.uk	USBCSW	
	Kishan	Patel	Kishan.Patel.5@city.ac.uk	USCSCI	
16	James	Mac	James.Mac@city.ac.uk	USCSCI	Abigail.Parisaca-Vargas.1@city.ac.uk
	Martin	Kaskatiyski	Martin.Kaskatiyski@city.ac.uk	USCSCI	
	Shilp	Patel	Shilp.Patel@city.ac.uk	USCSCI	
	Mohammed	Miah	Mohammed.Miah.7@city.ac.uk	USCSCI	
	Hamid	Ahmad	Hamid.Ahmad@city.ac.uk	USCSCS	
	Lasse	Marten	Lasse.Marten.2@city.ac.uk	USBCSW	
17	Mariia	Onyshchenko	Mariia.Onyshchenko@city.ac.uk	USGTEC	Abigail.Parisaca-Vargas.1@city.ac.uk
	Hamzah	Malik	Hamzah.Malik@city.ac.uk	USCSCI	
	Sharmistha	Dutt	Sharmistha.Dutt@city.ac.uk	USCSCI	

	Anthony	Gibson	Anthony.Gibson@city.ac.uk	USGTEC	
	Martin	Ivanov	Martin.Ivanov@city.ac.uk	USCSCS	
	Devina	Thakker	Devina.Thakker@city.ac.uk	USBCSW	
18	Ajwad	Mushtaq	Ajwad.Mushtaq@city.ac.uk	USCSCI	Abigail.Parisaca-Vargas.1@city.ac.uk
	Sumon	Miah	Sumon.Miah.2@city.ac.uk	USCSCI	
	Erikas	Dilys	Erikas.Dilys@city.ac.uk	USGTEC	
	Rahul	Sharma	Rahul.Sharma@city.ac.uk	USGTEC	
	Gurdeet	Singh	Gurdeet.Singh@city.ac.uk	USBCSW	
	Timothy	Shen	Timothy.Shen@city.ac.uk	USGTEC	
19	Sarath	Vaman	Sarath.Vaman@city.ac.uk	USCSCI	Simon.Odense@city.ac.uk
	Sajawal	Kiani	Sajawal.Kiani@city.ac.uk	USCSCI	
	Saqib	Amir	Saqib.Amir@city.ac.uk	USCSCI	
	Sugan	Nathan	Sugan.Nathan@city.ac.uk	USCSCI	
	Saamran	Gul	Saamran.Gul@city.ac.uk	USBCSW	
	Connor	Toms	Connor.Toms@city.ac.uk	USGTEC	
20	Najam-Us-Samad	Anjum	Najam-Us-Samad.Anjum@city.ac.uk	USBCSW	Mithileysh.Sathiyanarayanan@city.ac.uk
	Drilon	Agolli	Drilon.Agolli@city.ac.uk	USGTEC	
	Lehan	Ahmed	Lehan.Ahmed@city.ac.uk	USBCSW	
	Pascoal	Coutinho	Pascoal.Coutinho@city.ac.uk	USCSCI	
	Akramul	Islam	Akramul.Islam@city.ac.uk	USCSCI	
	Timur	Jalilov	Timur.Jalilov@city.ac.uk	USCSCI	
21	Robert	Zsolt	Robert.Zsolt@city.ac.uk	USCSCS	Mithileysh.Sathiyanarayanan@city.ac.uk
	Levi	Dowe-Ranger	Levi.Dowe-Ranger@city.ac.uk	USCSCI	
	Firoza	Patel	Firoza.Patel@city.ac.uk	USBCSW	
	Toran	Kopuk	Toran.Kopuk@city.ac.uk	USCSCI	
	Esther-Shira	Gaffin	Esther-Shira.Gaffin@city.ac.uk	USCSCI	
	Achraf	Chairi	Achraf.Chairi@city.ac.uk	USCSCI	
22	Luke	Tyson	Luke.Tyson@city.ac.uk	USCSCI	Mithileysh.Sathiyanarayanan@city.ac.uk
	Kiranjit	Dogra Ferre	Kiranjit.Dogra-Ferre@city.ac.uk	USCSCS	
	Albin	Trstena	Albin.Trstena@city.ac.uk	USCSCS	
	Jean-Glody	Mayala	Jean-Glody.Mayala@city.ac.uk	USCSCI	
	Patrick	Fisher	Patrick.Fisher@city.ac.uk	USCSCI	
	Charlie	Atkins	Charlie.Atkins@city.ac.uk	USCSCI	
	Esrak	Islam	Esrak.Islam@city.ac.uk	USBCSW	
23	Raymun	Kohli	Raymun.Kohli@city.ac.uk	USCSCI	Mithileysh.Sathiyanarayanan@city.ac.uk
	Mayur	Depala	Mayur.Depala@city.ac.uk	USCSCI	
	Anil	Yagliyurt	Anil.Yagliyurt@city.ac.uk	USBCSW	
	Parmveer	Johal	Parmveer.Johal@city.ac.uk	USCOSC	
	Myan	Patel	Myan.Patel@city.ac.uk	USCSCI	
	Zen	Khan	Zen.Khan@city.ac.uk	USCSCI	
	Lukas	Adomenas	Lukas.Adomenas@city.ac.uk	USCSCI	
24	Mihnea	Moldoveanu	Mihnea.Moldoveanu@city.ac.uk	USCSCI	Juan.Armond.2@city.ac.uk
	Nadeem	Hussain	Nadeem.Hussain.2@city.ac.uk	USBCSW	
	Sajjad	Razvi	Sajjad.Razvi@city.ac.uk	USCSCI	
	Saif	Rehman	Saif.Rehman@city.ac.uk	USCSCI	
	Mohammad	Farooq	Mohammad.Farooq@city.ac.uk	USGTEC	
	Kyle	Walke	Kyle.Walke@city.ac.uk	USCSCI	
	Christopher	Driver	Christopher.Driver@city.ac.uk	USCSCI	
25	Michal	Szpila	Michal.Szpila@city.ac.uk	USCOSC	Juan.Armond.2@city.ac.uk
	Trevor	Oluotch	Trevor.Oluotch@city.ac.uk	USCSCI	
	Daniel	Mckinley	Daniel.Mckinley@city.ac.uk	USGTEC	
	Mohammad	Ahmed	Mohammad.Ahmed.2@city.ac.uk	USCSCI	

	Sylvester	Mbagwu	Sylvester.Mbagwu@city.ac.uk	USCSCS	
	Sheena	Mistry	Sheena.Mistry@city.ac.uk	USCOSC	
	Shiven	Senthilkumar	Shiven.Senthilkumar@city.ac.uk	USBCSW	
26	Yasin	Syed	Yasin.Syed@city.ac.uk	USGTEC	Radha.Kopparti@city.ac.uk
	Declan	Walsh	Declan.Walsh@city.ac.uk	USCSCI	
	Ana-Maria	David	Ana-Maria.David@city.ac.uk	USBCSW	
	Asbin	Rai	Asbin.Rai@city.ac.uk	USCSCI	
	Samuel	Chan	Samuel.Chan@city.ac.uk	USGTEC	
	Aman	Patti	Aman.Patti@city.ac.uk	USBCSW	
	Uthmaan	Bhula	Uthmaan.Bhula@city.ac.uk	USCSCI	