UG Stage 2 Team Project (IN2018) module
2020/2021
All BSc and MSci students in the second year of their studies, excluding students on Professional Pathway

Developing a Software Product

BAPERS: Bloomsbury's Automated Process Execution Recording System

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.

Version 1.3
Jan. 2021

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 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. 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)
- 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*, 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/2.3/
    - 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 7 (+/- 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). Assignment 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, C#, JavaScript, 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
- Nilesh: designer/tester
- Anna: system analyst/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*** (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, Bloomsbury's Image processing Laboratory (BIPL). 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 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. 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 supervisors **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/supervisor – the latter type of meetings – *weekly* online via Teams or Zoom. Please note that team consultants are very busy (they are 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 team he/she supervises 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 (18 January – 2 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. 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	**20 January:** Briefing on module organisation (an extra lecture in week 1) **21 January:** 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	**21 January:** Interviews **28 January:** Briefing on System Design and Q&A session	Interviews with customer - further requirements elicitation. Students **must** attend the respective interview. See the schedule on Moodle	None

1–5	**04 February:** Database modelling recap **11 February:** Java GUI recap and Version Control Guest Lecture/Tutorial **18 February:** Briefing on System Implementation **Sun. 21 February, 5pm** (end of week 5)**:** *Submission* of: “*Requirements Specification and System Design*” document, and Project Binder and Individual Diaries.	Specifying system requirements and modelling system design. Project Binder and Individual Diaries maintenance.	“*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	**4 March:** General Feedback on the first deliverable, and Q&A session. **11 March:** Briefing on Database Connectivity programming (JDBC) **18 March**: Guest talk on Agile Methodology and Test-Driven development, and Q&A session. **25 March:** Preparation for Software Demo, and Q&A session. **TBC: Week ending 2 April:** *Demonstration* of the software in operation – summative assessment (Thur. 1 April, Whole day); *Submission* of Implementation Reports (Sun. 4 April, 5pm); **TBC: Week ending 11 April:** *Submission* of Individual Reports; (Sun. 11 April, 5pm); **TBC: Mon. 12 April, 5pm** Final versions of Project Binder and Individual Diaries, and any final evidence about individual contribution by any of the team members!	System Implementation and Testing. Attendance to the Demo. Preparation and submission of Implementation Reports. Preparation and Submission of the Individual Reports Project Binder and Individual Diaries maintenance, and submission.	Formal, summative assessments: 1. *Working system demonstration (“Demo”)* 2. *Implementation Report* 3. *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 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 (19 January – Tue!) – Introduce Team Project and its structure, learning objectives, assessments, etc.
- Week 1 (21 January) – What is a requirements specification? Guest lecture on effective team-working.
- Week 2 (28 January) – What is a system design? Q&A session, too.
- Week 3 (04 February) – Database modelling recap
- Week 4 (11 February) – Version control and Java GUI recap
- Week 5 (18 February) – What is software implementation?
- Week 7 (04 March) – General Feedback on the first deliverable. Q&A session.
- Week 8 (11 March) – Java JDBC programming
- Week 9 (18 March) – Guest talk on Agile Methodology and Test-Driven development, and Q&A session.
- Week 10 (25 March) – Preparation for Software Demo, and Q&A session.
- Week 11 (01 April) – Software Demo assessment [*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, 12	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 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, 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 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 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://www.city.ac.uk/__data/assets/pdf_file/0009/365292/Assessment-and-Feedback-Policy-Senate-October-2016-2.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 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). 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 (2)

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

- 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 already). This should simply be in natural language, using your own words. Students should not, however, "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 new 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 worked-out UC examples from Tutorial 2 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, 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, the teams are expected to use Relational Database Management Server (RDBMS), not an OR, or an OO, DBMS. 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, 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") is planned to take place synchronously online in week 11. The details about the Demo event (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 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 *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:

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

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

Once a task is completed, the semi- or fully-completed job is placed in a given location 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, and removing or updating the existing tasks.

BIPL carries out *jobs* on behalf of its customers. Each job is given a unique identifier on receipt, and assigned *urgency level*. *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, tens or even 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 interact with **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 functionality:
BAP-ACCT *Accept job at reception*: Identify existing customer account, or create new customer account. Assign job number. Record deadline for completion and any special instructions. This functionality will be mainly

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

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 a job through laboratory*: Respond to enquiries from any computer terminal about status of a job, or of all jobs in progress, or of all jobs (including the completed ones). Update status of a given job in progress by recording completion of the current task and commencement of the 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.
No job cancellations, or changes to jobs once accepted, are to be implemented (jobs do not change after they have been accepted).

BAP-REPT *Producing various reports*. The following reports are required: i) Individual report for the jobs of 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.
All types of reports must be both viewable and printable. Also, it must be possible to generate all types of reports automatically (the frequency at which they are generated should be configurable), and on-demand, for an arbitrary time period.

BAP-PAYM *Payment processing*. 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 the type of payment need to be stored. If a card payment is successful, the following details are recorded too: card type, expiry date and the last 4 digits of the card used. Cash payments are recorded, too. Only payment in full is accepted (i.e. no partial payments are allowed).
Valued customers are allowed to pay by the 10th of each (subsequent) 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 should generate alert as pop-up window, warning at regular intervals of 15 minutes until the Office Manager acknowledges the receipt of the warning.
No refunds are to be implemented in BAPERS.

BAP-CUST *Valued customers* may be given a discount. The specific terms for giving a discount are defined in a *discount plan* associated with 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 on demand, e.g. for a given job. 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 frequency at which they are generated should be configurable). The following user roles are essential to be implemented: Office Manager, Shift Manager, Receptionist and Technician. This functionality will be available to Office Manager only.

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 ongoing tasks and thus 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 clearly *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 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.

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. The case studies for OOAD coursework(s) are different from BAPERS. You are NOT allowed to provide UC specifications for any of the worked-out UC examples from Tutorial 2 of OOAD (IN2013) module. Also, 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 12/01/2020**

Customer: ACC3005
Description of work:
Estimated time for collection: 4 pm, 13 January 2020

Description of work in progress:

Job	Price, £	Task	Department	Start Time	Time Taken	Completed by
J20934565	19.00	1	Copy Room	12:00	110 min	John Nash
	49.50	2	Development	12:30	55 min	Lee Hong
	6.00	3	Packing	13:20	10 min	Marina Scott
J95869663	19.00	1	Copy Room	12:20	120 min	John Nash
	80.00	4	Development	13:10	85 min	Lee Hong
	6.00	3	Packing	14:00	20 min	Julie Abbot
J67549887	19.00	1	Copy Room	12:55	1 h 40 min	John Nash
	110.30	5	Development	14:40	175 min	Lee Hong
	6.00	3	Packing	15:20	25 min	Julie Abbot
J76544899	8.30	6	Copy Room	14:40	70 min	John Nash
	110.30	5	Development	15:10	160 min	Lee Hong
	55.50	7	Finishing Room	15:35	45 min	Julie Abbot
J65884565	49.50	2	Development	12:20	65 min	Stewart Pask
	6.00	3	Packing	12:50	30 min	Julie Abbot
J7659998	80.00	4	Development	13:10	100 min	Stewart Pask
	6.00	3	Packing	13:35	30 min	Julie Abbot

Account Details

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

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

Discount Rate:

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

Customer Name:	InfoPharma Ltd
Account No:	ACC0003
Contact Name:	Mr Alex Wright
Address:	25, Bond Street, London WC1V 8LS
Phone:	0207 321 8001
Agreed Discount:	*Flexible (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 / 13/01/2020

Account: ACC0001

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

Job No: J12456554 Completed: 13 January 2020

Description of work:

Task IDs	Price (£)
1	19.00
2	49.50
4	80.00
3	6.00
Sub-Total	154.50
Discount agreed:	3%
	149.87
Total payable (VAT at 20%)	179.84

Make a payment within 30 days by cash or card.

9.2. By Mr Rick Evans, Office Manager

Individual Performance Report

Period: 13/01/2020 – 13/01/2020

Name	Task IDs	Department	Date	Start time	Time taken	Total
John Nash	1	Copy Room	13/01/2020	12:00	20 min	2 h 55 min
	1	Copy Room	13/01/2020	12:20	35 min	
	1	Copy Room	13/01/2020	12:55	1 h 40 min	
	1	Copy Room	13/01/2020	14:40	20 min	
Lee Hong	2	Development	13/01/2020	12:30	40 min	2 h 10 min
	4	Development	13/01/2020	13:10	40 min	
	5	Development	13/01/2020	14:40	30 min	
	5	Development	13/01/2020	15:10	20 min	
Julie Abbot	3	Packing	13/01/2020	14:00	10 min	1 h 05 min
	3	Packing	13/01/2020	15:20	10 min	
	7	Finishing Room	13/01/2020	15:35	25 min	
	3	Packing	13/01/2020	12:50	10 min	
	3	Packing	13/01/2020	13:35	10 min	
Marina Scott	3	Packing	13/01/2020	13:20	10 min	0 h 10 min
Stewart Pask	2	Development	13/01/2020	12:20	30 min	0 h 50 min
	4	Development	13/01/2020	13:10	20 min	
Total effort:						7 h 10 min

Summary Performance Report

Period: 13/01/2020 – 20/01/2020

Day shift 1 (5 am – 2:30 pm)

Date	Copy Room	Development	Finishing	Packing
13/01/2020	8 h 20 min	10 h 45 min	8 h 30 min	2 h 30 min
14/01/2020	10 h 20 min	11 h 30 min	10 h 40 min	1 h 30 min
15/01/2020	7 h 20 min	13 h 20 min	9 h 30 min	2 h 50 min
16/01/2020	5 h 10 min	15 h 20 min	6 h 20 min	2 h 20 min
17/01/2020	7 h 10 min	10 h 50 min	7 h 20 min	3 h 25 min
18/01/2020	9 h 25 min	14 h 30 min	9 h 20 min	2 h 50 min
19/01/2020	10 h 30 min	12 h 40 min	11 h 30 min	1 h 30 min
20/01/2020	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
13/01/2020	8 h 20 min	13 h 45 min	12 h 30 min	2 h 30 min
14/01/2020	12 h 20 min	15 h 30 min	6 h 40 min	1 h 30 min
15/01/2020	10 h 20 min	16 h 20 min	7 h 30 min	3 h 20 min
16/01/2020	6 h 10 min	17 h 10 min	5 h 20 min	2 h 20 min
17/01/2020	4 h 10 min	11 h 50 min	8 h 20 min	4 h 25 min
18/01/2020	13 h 25 min	10 h 30 min	9 h 20 min	2 h 50 min
19/01/2020	12 h 30 min	16 h 20 min	10 h 30 min	3 h 30 min
20/01/2020	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
13/01/2020	8 h 20 min	5 h 45 min	8 h 30 min	1 h 30 min
14/01/2020	6 h 20 min	7 h 30 min	6 h 40 min	0 h 30 min
15/01/2020	3 h 20 min	2 h 20 min	5 h 30 min	1 h 50 min
16/01/2020	2 h 10 min	4 h 20 min	3 h 20 min	1 h 20 min
17/01/2020	1 h 10 min	3 h 50 min	3 h 20 min	1 h 40 min
18/01/2020	7 h 25 min	5 h 30 min	4 h 20 min	1 h 30 min
19/01/2020	7 h 30 min	4 h 40 min	6 h 30 min	1 h 30 min
20/01/2020	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 (13/01/2020 – 20/01/2020)

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
Night 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-CUST and BAP-ADMN.
Receptionist: Access to BAP-ACCT, and can also record payments that have been made.
Technician: Access to BAP-PROC

10. Team membership

Team	First Name	Surname	Email ID	Consultant
1	DILLON	BHANDERI	Dillon.Bhanderi@city.ac.uk	Aravin.Naren.1@city.ac.uk
	ENI	ZEQO	Eni.Zeqo@city.ac.uk	
	HUZAINAH	SHAH	Huzainah.Shah@city.ac.uk	
	ILYAS	AYOUB-BEY	Ilyas.Ayoub-Bey@city.ac.uk	
	MOHAMMED	UDDIN	Mohammed.Uddin.11@city.ac.uk	
	MUHAMMAD	RAHMAN	Muhammad.Rahman.5@city.ac.uk	
	RHYS	WOOTTON	Rhys.Wootton@city.ac.uk	
2	ANDREYA	DINEV	Andreya.Dinev@city.ac.uk	Aravin.Naren.1@city.ac.uk
	DANISH	HUSSAIN	Danish.Hussain@city.ac.uk	
	EMMANUEL	MUKUNGWA	Emmanuel.Mukungwa@city.ac.uk	
	LEVI	BENOLIEL ELLIOTT	Levi.Benoliel-Elliott@city.ac.uk	
	MUHLIS	KARAKUS	Muhlis.Karakus@city.ac.uk	
	PRASHAN	RAI	Prashan.Rai@city.ac.uk	
	YAZAN	ALI	Yazan.Ali@city.ac.uk	
3	CHRISTOPHER	RAINBIRD	Christopher.Rainbird@city.ac.uk	Aravin.Naren.1@city.ac.uk
	JAY	TAURON	Jay.Tauron@city.ac.uk	
	LABINOT	BAJQINCA	Labinot.Bajqinca@city.ac.uk	
	MOHYIDIAN	MOHAMED	Mohyidian.Mohamed@city.ac.uk	
	RANIA	LAHLOU	Rania.Lahlou@city.ac.uk	
	VICTOR	BUDACA	Victor.Budaca@city.ac.uk	
	VICTORIA	BROOMFIELD	Victoria.Broomfield@city.ac.uk	
4	FLORA	DAS	Flora.Das@city.ac.uk	Charles.Watson.1@city.ac.uk
	HASSAN	FAROOQ	Hassan.Farooq.2@city.ac.uk	
	MOHAMMED	SYEDUL	Mohammed.Syedul@city.ac.uk	
	PRINCE	HUSSAIN	Prince.Hussain@city.ac.uk	
	SADIYAH	SAEED	Sadiyah.Saeed@city.ac.uk	
	SAHIB	CHOWDHURY	Sahib.Chowdhury@city.ac.uk	
	SYED	MIAH	Syed.Miah.2@city.ac.uk	
5	ASAD	ZAFAR	Asad.Zafar@city.ac.uk	Charles.Watson.1@city.ac.uk
	EFTEKHAR	UDDIN	Eftekhar.Uddin@city.ac.uk	
	ELYAS	NOUI	Elyas.Noui@city.ac.uk	
	INTHIRAJITH	SINNIAH	Inthirajith.Sinniah@city.ac.uk	
	MOEED	JAVAID	Moeed.Javaid@city.ac.uk	
	MOSES	OCHIENG	Moses.Ochieng@city.ac.uk	
	SUNIL	RAMJI BHANDERI	Sunil.Ramji-Bhanderi@city.ac.uk	
6	ARMET	GILL	Armet.Gill@city.ac.uk	Charles.Watson.1@city.ac.uk
	IKRAM	RASHID	Ikram.Rashid@city.ac.uk	
	JAVONNE	PORTER	Javonne.Porter@city.ac.uk	
	MUHAMMAD	SOHAIL	Muhammad.Sohail@city.ac.uk	
	OZGUR	CAKICI	Ozgur.Cakici@city.ac.uk	
	RYO	MOCHIZUKI	Ryo.Mochizuki@city.ac.uk	
	VALLIKKANNU	RAMASWAMY	Vallikkannu.Ramaswamy@city.ac.uk	
7	ABDULLAH	KHAN	Abdullah.Khan.3@city.ac.uk	Marius.Zicius.2@city.ac.uk
	FARHAN	HABIBIE	Farhan.Habibie@city.ac.uk	
	HANAN	BUDRAA	Hanan.Budraa@city.ac.uk	
	MUNISH	PEENITH	Munish.Peenith@city.ac.uk	
	PETER	FARKAS	Peter.Farkas@city.ac.uk	
	RASHIDUL	RAHIN	Rashidul.Rahin@city.ac.uk	

	TULSI	AMIN	Tulsi.Amin@city.ac.uk	
8	BABATUNDE	ADESOYE	Babatunde.Adesoye@city.ac.uk	Marius.Zicius.2@city.ac.uk
	FAIZA	SIDDIQUI	Faiza.Siddiqui@city.ac.uk	
	HAMZA	KHAN	Hamza.Khan.4@city.ac.uk	
	MEHEDI	SARDAR	Mehedi.Sardar@city.ac.uk	
	MUHAMMAD	KHAN	Muhammad.Khan.30@city.ac.uk	
	NICHOLAS	PREKDUCA	Nicholas.Prekduca@city.ac.uk	
	SUMAYYAH	ISLAM	Sumayyah.Islam@city.ac.uk	
9	ABDULAZIZ	ALSHAFEI	Abdulaziz.Alshafei@city.ac.uk	Marius.Zicius.2@city.ac.uk
	ABUL	CHOWDHURY	Abul.Chowdhury@city.ac.uk	
	AIDAN	WOOD	Aidan.Wood@city.ac.uk	
	ALEJANDRO	GARCIA PACHECO	Alejandro.Garcia-Pacheco@city.ac.uk	
	AMY	WRIGHT	Amy.Wright.3@city.ac.uk	
	DAWUD	MARIKAR	Dawud.Marikar@city.ac.uk	
	SHUKRI	AHMED	Shukri.Ahmed@city.ac.uk	
10	AMIR	HARVEY	Amir.Harvey@city.ac.uk	K.Pal@city.ac.uk
	AYUSHI	BIMALI	Ayushi.Bimali@city.ac.uk	
	BILAL	HASSAN	Bilal.Hassan.2@city.ac.uk	
	JUSTIN	DOMINGUES	Justin.Domingues@city.ac.uk	
	LUCAS	WILSON	Lucas.Wilson@city.ac.uk	
	VINUSHAN	JEYAKUMAR	Vinushan.Jeyakumar@city.ac.uk	
	YAIRO	BECERRA GUTIERREZ	Yairo.Becerra-Gutierrez@city.ac.uk	
11	ABARNA	AHILAN	Abarna.Ahilan@city.ac.uk	K.Pal@city.ac.uk
	AMMEL	OSMAN	Ammel.Osman@city.ac.uk	
	ARHUM	ASHRAF	Arhum.Ashraf@city.ac.uk	
	AYMAN	UDDIN	Ayman.Uddin@city.ac.uk	
	EDWARD	LEONARD	Edward.Leonard@city.ac.uk	
	KACPER	ZOLTOWSKI	Kacper.Zoltowski@city.ac.uk	
	MOHAMMED	UDDIN	Mohammed.Uddin.13@city.ac.uk	
12	ABIGAIN	MAMANI PINTO	Abigain.Mamani-Pinto@city.ac.uk	K.Pal@city.ac.uk
	AHMED	OSMAN	Ahmed.Osman@city.ac.uk	
	DHRUV	PATEL	Dhruv.Patel@city.ac.uk	
	MARKO	KOVAC	Marko.Kovac@city.ac.uk	
	NIAN	BU	Nian.Bu@city.ac.uk	
	TAYYAB	HUSSAIN	Tayyab.Hussain@city.ac.uk	
	WIKTOR	WIELENTO	Wiktor.Wielento@city.ac.uk	
13	ADNAAN	MOHAMOUD	Adnaan.Mohamoud@city.ac.uk	Abigail.Parisaca-Vargas.1@city.ac.uk
	ANDREEA-DANIELA	BACIU	Andreea-Daniela.Baciu@city.ac.uk	
	BENJAMIN	YIP	Benjamin.Yip@city.ac.uk	
	MOHAMED	BUSSURI	Mohamed.Bussuri@city.ac.uk	
	RIMU	AKTHER	Rimu.Akther@city.ac.uk	
	SAHIB	BHATTI	Sahib.Bhatti@city.ac.uk	
	VIRALI	SHAH	Virali.Shah.2@city.ac.uk	
14	ADAM	KHANZADA	Adam.Khanzada@city.ac.uk	Abigail.Parisaca-Vargas.1@city.ac.uk
	ANNA	KANTORCZYK	Anna.Kantorczyk@city.ac.uk	
	HANNAH	THOMAS	Hannah.Thomas.2@city.ac.uk	
	MUHAMMAD	ABDULLAH	Muhammad.Abdullah.3@city.ac.uk	
	OSMAN	MURAD	Osman.Murad@city.ac.uk	
	STEFAN	UZUNOV	Stefan.Uzunov@city.ac.uk	
	TOMASZ	TLALKA	Tomasz.Tlalka@city.ac.uk	
15	ARCHIE	SHORT	Archie.Short@city.ac.uk	Abigail.Parisaca-

	MATTHEW	CLOKE	Matthew.Cloke@city.ac.uk	Vargas.1@city.ac.uk
	SHABAN	HASAN	Shaban.Hasan@city.ac.uk	
	SHAHBAZ	SYED	Shahbaz.Syed@city.ac.uk	
	WILLIAM	DIBBLE	William.Dibble@city.ac.uk	
	YAHYE	OSMAN	Yahye.Osman@city.ac.uk	
	YASH	THAKER	Yash.Thaker@city.ac.uk	
16	ASKARI	SHAH	Askari.Shah@city.ac.uk	Abigail.Parisaca-Vargas.1@city.ac.uk
	IGOR	PEREIRA-MARTINS	Igor.Pereira-Martins@city.ac.uk	
	KALOYAN	DIMITROV	Kaloyan.Dimitrov@city.ac.uk	
	NOEL	QUADRI	Noel.Quadri@city.ac.uk	
	RADU	TAICA	Radu.Taica@city.ac.uk	
	SAMIL	AHMED	Samil.Ahmed@city.ac.uk	
	SUFYAAN	AMNOUR	Sufyaan.Amnour@city.ac.uk	
17	GIRUDHIHAN	MOHANARAJAH	Girudhihan.Mohanarajah@city.ac.uk	Abigail.Parisaca-Vargas.1@city.ac.uk
	KRZYSZTOF	SZYMANSKI	Krzysztof.Szymanski@city.ac.uk	
	MARK	CARTER	Mark.Carter@city.ac.uk	
	MOHAMAD	AL SOUFI	Mohamad.Alsoufi@city.ac.uk	
	MOHAMMED	MIAH	Mohammed.Miah.16@city.ac.uk	
	SHAJEEVAN	CHANDRASEKARAN	Shajeevan.Chandrasekaran@city.ac.uk	
	YASSIN	KERBOUBA	Yassin.Kerbouba@city.ac.uk	
18	DANIEL	SZABO	Daniel.Szabo@city.ac.uk	Beatrice.Vincenzi@city.ac.uk
	JOE	DAVIS	Joe.Davis@city.ac.uk	
	ROBERT	KUTTA	Robert.Kutta@city.ac.uk	
	SAFEE	MAHOMADE	Safee.Mahomade@city.ac.uk	
	SAYEED	AHMED	Sayeed.Ahmed@city.ac.uk	
	TAYO	AJIBODE-FOLKES	Tayo.Ajibode-Folkes@city.ac.uk	
	WALEE	AHMED	Walee.Ahmed@city.ac.uk	
19	ABDULLAH	KHAN	Abdullah.Khan.2@city.ac.uk	Beatrice.Vincenzi@city.ac.uk
	ADAM	BARRE	Adam.Barre@city.ac.uk	
	BAILEY	WIFFEN	Bailey.Wiffen@city.ac.uk	
	DANIIL	CHERKASOV	Daniil.Cherkasov@city.ac.uk	
	FINLAY	QUINTON	Finlay.Quinton@city.ac.uk	
	RAED	IFTIKHAR	Raed.Iftikhar@city.ac.uk	
	ZOHAIB	SALAM	Zohaib.Salam@city.ac.uk	
20	ADAM	MOHSIN	Adam.Mohsin@city.ac.uk	Beatrice.Vincenzi@city.ac.uk
	DAHIR	NOOR	Dahir.Noor@city.ac.uk	
	JACOB	JOHNSON	Jacob.Johnson@city.ac.uk	
	JIMM	INCIONG	Jimm.Inciong@city.ac.uk	
	LUKE	SIMPSON	Luke.Simpson@city.ac.uk	
	NASIM	CHOUDHURY	Nasim.Choudhury@city.ac.uk	
	RUI	YANG	Rui.Yang@city.ac.uk	
21	ALI	ATAYA	Ali.Ataya@cass.city.ac.uk	Beatrice.Vincenzi@city.ac.uk
	GERA	JAHJA	Gera.Jahja@city.ac.uk	
	JACOB	RODELL	Jacob.Rodell@city.ac.uk	
	KEHKSHAN	BHATTI	Kehkshan.Bhatti@city.ac.uk	
	MANPREET	KAUR	Manpreet.Kaur.4@city.ac.uk	
	MUHAMMAD	ANSARI	Muhammad.Ansari@city.ac.uk	
	MUSHFIKUR	RAHMAN	Mushfikur.Rahman@city.ac.uk	
22	ARUNAS	PAKALNISKIS	Arunas.Pakalniskis@city.ac.uk	Damien.Fernando@city.ac.uk
	AVROHOM	DOWEK	Avrohom.Dowek@city.ac.uk	
	JAKUB	ZAVACKY	Jakub.Zavacky@city.ac.uk	
	NATHAN	ODIBO	Nathan.Odibo@city.ac.uk	

	SEVER	MIRANBEG	Sever.Miranbeg@city.ac.uk	
	SUBEKA	PREMACHANDRAN	Subeka.Premachandran@city.ac.uk	
	TAUHID	HOSSAIN	Tauhid.Hossain@city.ac.uk	
23	ALIA	ZIANI	Alia.Ziani@city.ac.uk	Damien.Fernando@city.ac.uk
	AMEER	ABDEL-HUSSEIN	Ameer.Abdel-Hussein@city.ac.uk	
	HAROON	AHMED	Haroon.Ahmed@city.ac.uk	
	ISHRAT	AHMED	Ishrat.Ahmed@city.ac.uk	
	JACK	DEZITTER	Jack.Dezitter@city.ac.uk	
	MOHAMMED	KHAN	Mohammed.Khan.19@city.ac.uk	
	ZOHAIB	AHMED	Zohaib.Ahmed@city.ac.uk	
24	ALIA	REZVI	Alia.Rezvi@city.ac.uk	Damien.Fernando@city.ac.uk
	FAIZAAN	ADIA	Faizaan.Adia@city.ac.uk	
	MAHISHA	KHAN	Mahisha.Khan@city.ac.uk	
	MOHAMED	SHIRE	Mohamed.Shire@city.ac.uk	
	OLIVER	REEKIE	Oliver.Reekie@city.ac.uk	
	PAUL	TARANU	Paul.Taranu@city.ac.uk	
	RAHIM	AHMED	Rahim.Ahmed@city.ac.uk	
25	ADRIANNA	BUDZIK	Adrianna.Budzik@city.ac.uk	Damien.Fernando@city.ac.uk
	MOEEN	YOUNIS	Moeen.Younis@city.ac.uk	
	MOHAMMAD	RAHMAN	Mohammad.Rahman.8@city.ac.uk	
	MOHAMMED	HUSSAIN	Mohammed.Hussain.19@city.ac.uk	
	ROSEN	GEORGIEV	Rosen.Georgiev@city.ac.uk	
	UMER	BUTT	Umer.Butt@city.ac.uk	
	CHARLES	ODILI	Charles.Odili@city.ac.uk	
26	AMIRALI	TORABI	Amirali.Torabi@city.ac.uk	Kaleem.Peeroo@city.ac.uk
	KINZA	BUTT	Kinza.Butt@city.ac.uk	
	MOHAMMED	TAHAR	Mohammed.Tahar@city.ac.uk	
	REZA	MAHMOOD	Reza.Mahmood@city.ac.uk	
	SAMUEL	KAUFMAN	Samuel.Kaufman@city.ac.uk	
	THISSHORE	KANTHARUBAN	Thisshore.Kantharuban@city.ac.uk	
	YUSUP	MYRATDURDYYEV	Yusup.Myratdurdyyev@city.ac.uk	
27	ENGLI	MARTINAJ KRUE	Engli.Martinaj-Krueziu@city.ac.uk	Kaleem.Peeroo@city.ac.uk
	IRINA-MARIA	FRATILA	Irina-Maria.Fratila@city.ac.uk	
	MUHAMMAD	MIAH	Muhammad.Miah.6@city.ac.uk	
	NICHOLAS	ELDERGILL	Nicholas.Eldergill@city.ac.uk	
	OLUWATOBI	ADEWUNMI	Oluwatobi.Adewunmi@city.ac.uk	
	OMAR	KHODABAKSH	Omar.Khodabaksh@city.ac.uk	
	SUBHAAN	MAJID	Subhaan.Majid@city.ac.uk	
28	DAN-ANDREI	COLTAN	Dan-Andrei.Coltan@city.ac.uk	Kaleem.Peeroo@city.ac.uk
	MOUSUF	ZAMAN	Mousuf-C-Zaman.Zaman@city.ac.uk	
	MUJAHID	AHMED	Mujahid.Ahmed@city.ac.uk	
	NAZIF	MOHAMED HANIF	Nazif.Mohamed-Hanif@city.ac.uk	
	RAUF	GANJIYEV	Rauf.Ganjiyev@city.ac.uk	
	TAYLOR	SQUIBB	Taylor.Squibb@city.ac.uk	
	VARUNINDRA	LAL	Varunindra.Lal@city.ac.uk	
29	MASHAL	ZAHRA	Mashal.Zahra@city.ac.uk	Kaleem.Peeroo@city.ac.uk
	SAFAT	SIKDER	Safat.Sikder@city.ac.uk	
	SHAUN	SEVUME	Shaun.Sevume@city.ac.uk	
	THANIM	HASAN	Thanim.Hasan@city.ac.uk	
	VIJAY	KESIREDDY	Vijay.Kesireddy@city.ac.uk	
	ZAID	ASHRAF	Zaid.Ashraf@city.ac.uk	
	ZISHAN	WANG	Zishan.Wang@city.ac.uk	

30	ABDULLAH	AL-SAADOUN	Abdullah.Al-Saadoun@city.ac.uk	Pedro.Magalhaes-Marques@city.ac.uk
	HOANG	NGUYEN KIM	Hoang.Nguyen-Kim@city.ac.uk	
	JOEL	GOGO	Joel.Gogo@city.ac.uk	
	PIOTR	PREUSCHOFF	Piotr.Preuschoff@city.ac.uk	
	SAMIN	TORABI	Samin.Torabi@city.ac.uk	
	XAVIER	MURROW	Xavier.Murrow@city.ac.uk	
31	ANDREA	CORTES	Andrea.Cortes@city.ac.uk	Pedro.Magalhaes-Marques@city.ac.uk
	FARHAN	AHMAD	Farhan.Ahmad@city.ac.uk	
	HAMZAH	HAMAYOUN	Hamzah.Hamayoun@city.ac.uk	
	MARTIN	BOTEV	Martin.Botev@city.ac.uk	
	NABIL	AHMED RUHIT	Nabil.Ahmed-Ruhit@city.ac.uk	
	SOPHIE	GUELI	Sophie.Gueli@city.ac.uk	
32	AATIQAH	DIWAN	Aatiqah.Diwan@city.ac.uk	Pedro.Magalhaes-Marques@city.ac.uk
	DINOSHAN	ARULANANTHAM	Dinoshan.Arulanantham@city.ac.uk	
	MIRELLA	MEDEIROS MELO	Mirella.Medeiros-Melo@city.ac.uk	
	MUHAMMAD	ANWAR	Muhammad.Anwar.5@city.ac.uk	
	SHIVEN	SAINI	Shiven.Saini@city.ac.uk	
	ZHE	CHONG	Zhe.Chong@city.ac.uk	
33	DARDAN	GASHI	Dardan.Gashi@city.ac.uk	Pedro.Magalhaes-Marques@city.ac.uk
	JONATHAN	SPINA ARENAS	Jonathan.Spina-Arenas@city.ac.uk	
	KHALED	ELMALLAH	Khaled.Elmallah@city.ac.uk	
	MEELAD	MUJADIDI	Meelad.Mujadidi@city.ac.uk	
	MUHAMMED	HASAN	Muhammed.Hasan@city.ac.uk	
	OLEKSANDRA	SOKOLAN	Oleksandra.Sokolan@city.ac.uk	