Queen Mary
University of London

School of Electronic Engineering and Computer Science

EBU6304 – Software Engineering Group Project

30% coursework.

A learning journey application

-developing the software using Agile Methods

1. General information

In the next few weeks, your team will be required to develop a Learning Journey Application using **Agile** methods. Your team should aim to deliver a simple first release of the software product that could be extended in further iterations. Agile methods should be used in all activities, from requirements, through to analysis/design, implementation and testing. Iterations should be planned and outcomes should be submitted.

There is no restriction on what the Learning Journey Application should include and the given specification contains only high-level abstract requirements. It should be noted that determining the software requirements is one of the most important and complex phases in any development project. You should apply requirement finding techniques and Agile methods to identify the actual requirements at an appropriate level. *Most importantly, you need to prioritise the features that are implemented in accordance with both ease of implementation and meeting requirements*. As in real software though, you should define the project SCOPE properly. Keep your design SIMPLE. Bear in mind that there is no absolute right answer – your solution may be perfectly appropriate.

Handout release date: **13th March 2023**
Final QMPlus submission (Product backlog, Prototype, Report and Software): **26th May 2023**
Demonstration: **27th May – 4th June 2023**
Marks returned: Approximately 2-3 weeks after the final submission.

2. Specification of the project

2.1 Basic requirements

Your team will develop a Learning Journey Application for BUPT International School students. The aim of this application is to help students record and track their learning journey over the 4 years study. There is no restriction on what functions the Learning Journey Application should have, as one of the most important tasks of this project is to identify the real requirements. These are some suggested functions to help you get started.

- Record modules studied
- Record skills gained
- Record marks achieved and produce relevant calculations for various purpose (e.g. degree classification, GPA etc)
- Record achievements (e.g. awards, prizes and honours etc)
- Roles undertaken (e.g. class rep, module rep and volunteers etc)
- Extra curriculum performed (e.g. research and project that is extra to the standard curriculum)
- Import/export of data
- Portfolios (outcomes produced, e.g. posters, videos, software etc)
- Any other function(s) that is useful

A full prototype of the application should be produced. It is not required to implement the full working code however your team should implement at least **THREE** core functions of your choice.

2.2 Other requirement

- The software must be developed using Java as a stand-alone application running on computers. A simple graphic user interface (GUI) should be used. The recent Java Edition should be used. Do NOT build a Web-based application or Phone App.
- The application should be used without an Internet connection.
- All input and output data should be in simple text file format. You may use plain text (txt), CSV, JSON, or XML. Do NOT use a database.
- Basic restrictions and error checking must be considered.
- Your design must be flexible and extensible to adapt to future changes: it should be easy to add new functions and can be extended for other user groups. When doing so, you should be able to reuse the existing components and make the least impact on the existing code.

Your tasks are to define detailed requirements, design, develop and test the above described software using Agile methods. Feel free to design the software as long as it satisfies the basic requirements, define the **SCOPE** properly.

3. Agile project management

Each group has 6 students. You are the Agile team working together to complete the project. All students in a group must work on all aspects of the project, to obtain full software engineering skills. You should use the techniques you have learnt in the lectures to manage the project, e.g., Scrum, daily stand up meetings, working around a table, scrum master and decision making, etc. You are also encouraged to use other efficient ways of communication to coordinate the group activities.

Suggested Timeline:

- 13-15 March: Meet group members, appoint a group leader and discuss the project handout.
- 16-24 March: Gather real requirements. Story writing workshop. Outcomes: product backlog and prototype.
- 27-31 March: Iteration 1. Outcomes: Working Software v1.
- 3-14 April: Iteration 2. Outcomes: Working Software v2.
- 17-28 April: Iteration 3. Outcomes: Working Software v3.
- 1-12 May: Iteration 4. Outcomes: Working Software v4.
- 15-26 May: Software final delivery.

4. Final submission on QM+: 26th May 2023

The final submission includes **product backlog, full prototype, a short report, and software**. **For all the submissions, only the group leader should submit the files on behalf of the whole group.**

4.1 The product backlog, an excel file (refer to the template on QM+). Filename: Productbacklog_groupXXX.xlsx, where XXX is your group number. It should contain all user stories with priority, estimation and iteration plan.

4.2 The prototype, a PDF file. Filename: Prototype_groupXXX.xlsx, where XXX is your group number. It should contain full prototype. Only low-fidelity prototype is needed.

4.3 The short report (maximum 15 pages**),** a PDF file. Filename: Report_groupXXX.xlsx, where XXX is your group number.

It should contain the following parts:

a. The purpose and scope of the application. E.g. define users and describe main features.
b. Project management in your team working. E.g., describe techniques, tools, planning, estimating, decision making and adapting to changes.
c. Requirements gathering. E.g. describe the use of requirements finding techniques. Describe iteration, prioritisation and estimation methods.

d. Analysis and Design. A set of design class diagrams describing the design of the software, show the class relationships. Explain the design of the application in various levels, e.g. architecture, classes, components, design patterns and design principles.

e. Implementation and Testing. Discuss the implementation strategy and iteration/built plan. Discuss the test strategy and test techniques you have used in your testing. Discuss the using of TDD. Note: TDD is not required for developing the whole software, however, you should try to use TDD to develop a few programs.

f. Next iterations (future): describe the next iterations of the software in the future.

The short report should have maximum 15 pages. Cover page, table of contents, appendices, reference list are EXCLUDED from page length. Tables, figures, diagrams, charts in the report main body are INCLUDED in page length.

4.4 The software, a ZIP file. Filename: Software_groupXXX.xlsx, where XXX is your group number.

It should contain the following parts:

a. Java code. THREE core functions should be implemented. Code should be well documented.

b. A set of test programs using Junit as an example of using TDD.

c. JavaDocs.

d. A user manual with some key screenshots of the application.

e. A readme file to instruct how to set up or configure and run your software.

5. Demonstration 27th May – 4th June 2023

Your team is required to demonstrate of your software product. ALL group members MUST attend the demonstration session. Detailed time and instruction of demonstration will be sent out in due course. You should demonstrate the running of your software product, as if you do the demonstration to sell your product. You should demonstrate the operation of the THREE core functions. It's recommended to demonstrate error handling as well. You need to answer a few questions.

6. The role of Teaching Assistants

Each group will be assigned a Teaching Assistant (TA) to provide support, feedback and monitor the group progress. Your TA should be your first contact if you have questions or issues. The TAs will regularly check your group progress and individual contribution.

7. Marks breakdown (approximate)

Group mark (maximum 100 marks)

Requirements: 30%

- Ability to extract and define the software requirements using Agile techniques. Use of appropriate fact-finding techniques. The correctness of defining scope and roles. The correctness of writing user stories. Correctness and completeness of product backlog. Quality of prototype.

Analysis and design: 20%

- Ability to refine the requirements through analysis. Ability to design high-quality software. Quality of the design class diagrams.

Implementation and testing: 20%

- The correctness of Java code. Quality of code. Appropriate test strategy. Unit testing. Integration testing.

Project management: 10%

- Appropriate use of tools for project management and communication. Appropriate use of project management techniques. Evidence of progress throughout the project period.

Report: 10%

- Quality of report writing.

Demonstration: 10%

- Demonstrate the software working correctly as intended.

Individual mark

Individual marks will be given according to the participation in the group: Quality of work performed and Understanding of the performed work. Each student will be evaluated through the evidence of contribution. The grade will be awarded as below:

A+	Outstanding	Receive 100% group marks + a maximum 5% extra
A	Satisfactory	Receive 100% group marks
B	Unsatisfactory	Receive 50% of group marks
C	No contribution	Receive 0% of group marks

You, AS A GROUP, are responsible for managing any issues and for completing all of the tasks. If students are not contributing to the group work, then the module organiser needs to be informed immediately.

Please use the student forum on QMPlus for general enquires and discussions.