Project Title: Student Tool for Reinforcing the Intake of Valuable Experience (STRIVE)

Research Team:

Student Partners

Student partner 1: Carla Garrido Augusto
Student number: w1570462
Email: w1570462@my.westminster.ac.uk
year/level of study: BENG (Hons) Software Engineering, Level 6

Student partner 2: Mohammad Momeni
Student number: w1638831
Email: w1638831@my.westminster.ac.uk
Course and year/level of study: BENG (Hons) Software Engineering, Level 5

Student partner 3: Benjamin Taylor
Student number: w1667280
Email: w1667280@my.westminster.ac.uk
Course and year/level of study: BENG (Hons) Software Engineering, Level 5

Student partner 4: Peter Maeder
Student number: w1604440
Course and year/level of study: BSC (Hons) Computer Science, Level 5

Academic Partners

Academic partner 1: Dr Alexander Bolotov
Email: A.Bolotov@wmin.ac.uk

Academic partner 2: Dr Gabriele Pierantoni
Email: g.pierantoni@westminster.ac.uk

Academic partner 3: David Chan
Email: davidchan@luminalearning.com, d.chanyoufee@westminster.ac.uk

College/School: Design, Creative and Digital Industries/ School of Computer Science and Engineering

Academic Year 2018-2019

A STUDENTS AS CO-CREATORS PROJECT
UNIVERSITY OF WESTMINSTER

Content:

List of Figures

Section 1. Executive Summary (300-400 words)

The project talked the improvement of the existing education interactive software EnAbled which was developed within the QHT funded project in 2018. The project team worked in two directions:

1. Improvement of the EnAbled functionality by linking the modules.
2. Enhancing EnAbled with the assessment functionality.

Work in the first direction researched the ways to implement a system which allows related/ pre-requisite module to be linked together, as well as depicting it using the vis graphic application.
As a result, the interface prototype of a dependent dynamic dropdown on the module edit page was developed. This would allow one to pick a module, then a topic of choice to link to the selected module.

Work in the second direction tackled the link between EnAbled and the assessments: after the analysis of the requirements of the selected module (5COSC004W Client Server Architecture) had been performed, two fundamental functionalities of great usefulness to both students and members of the module team were formulated.

a. An interactive environment to learn how to:
    i. Write a Client from an Existing Server
    ii. Write both a Client and a Server from specifications

    and an interactive environment to test the code implemented in the point (a)
b. An automated grading system that supports the members team in grading large numbers of exams that consist in code developed in point (a). Each student had to produce code to cover both (i) and (ii) and the grading work was significant given that the number of students was around 200. That meant 400 exams (i and ii), each consisting of several files. That means grading thousands of files.

The team has explored various methodologies – research methodologies related to software development, software development methodologies themselves. The student team should have initially gained sufficient knowledge on the existing software, its architecture and specifics of the implementation. Subsequently, the work followed the software development research methodology. Additionally, as of the important tasks was to grasp th existing and functioning software, the reverse engineering approach, widely used in this area, was explored.
The work on the software development was guided by the software development methodology. Much of it was related to the findings of the ways to add the functionality to the existing software which gave the team ideas of th functional requirements of the desired software to meet. The design stage involved modelling together with the choice of programming solutions. This mainly related to the work on linking EnAbled with the assessments. For th programming solutions of this scale, it was necessary to come up with good understanding of the software architecture which was another part of the software development methodology. Students gained important knowledge for the students involved into project was to carry on what Is called proof of concept of the architectur In other words, all pieces of the system design should have been tested to be able to “talk to each other”.
The feedback we gained was enriched by involving the students of the foundation year at our partner institution – WIUT – when the software was offered to them for the revision on Mathematics module in November 2018.
This has not been an easy journey, the project has been of a large scale and the team learned lessons. This repo reflects both our achievements and these lessons.

Tip: it might be a good idea to write this section last, although it needs to be at the beginning of your report.

Section 2. Background and Aims (200-300 words)

Strive is a co-creation project that allowed students to build personalized content used the EnAbled platformю EnAbled is an interactive teaching and learning software, the result of a Quintin Hogg funded project led by Dr. Alexander Bolotov, and jointly supervised by Dr A. Bolotov and Dr G. Pierantoni. EnAbled offers a "digital map and compass" to successfully navigate through the "academic maze". Its underlying assumption are that: nowadays students have access to a vast and heterogeneous amount of learning resources (from traditional books, to videos and e-learning platforms, etc…), and, that students have their own personal learning style (which reflects their personality) and that such styles may be very different. The combination of these two facts gives rise to a great variety of different learning paths that students follow in their University years.
On the other hand, it is worth noticing that when Module Leaders design their learning material have their own optimal "learning path" in mind and that this may not necessarily coincide with that of some students. EnAbled tries to build a bridge that brings together the perspective of the student and that of the module leader. EnAbled allows module leaders to model different learning paths, each suited for a different student personality. As an example, the map in Figure 1 shows a prototype of the overall structure of the Computer Science Module Client/Server Architecture modelled as an oriented graph.

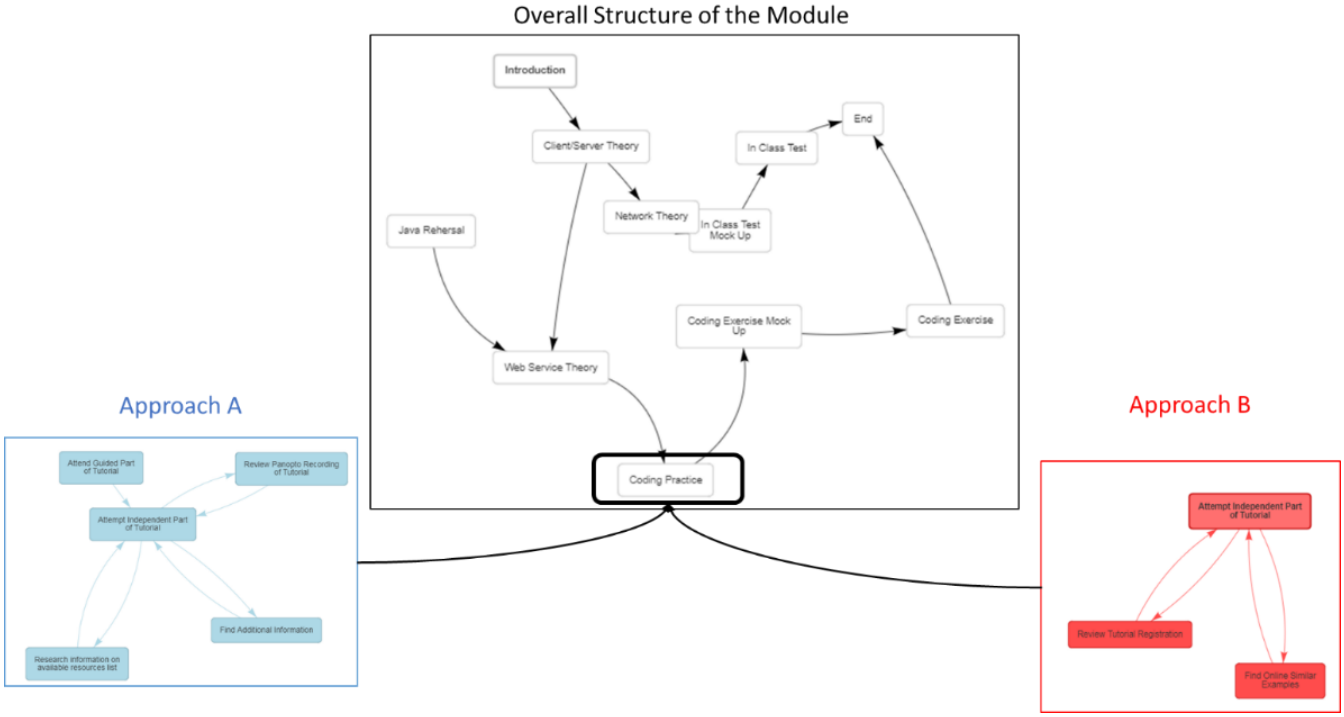

Figure 1, Module Leader View

It also offers different approaches (coded as coloured sub-graphs) for each step. As an example, the step Coding Practice has two possible approaches (A - in blue, and B – in red). Approach A is for the student who prefers to

follow the guided part of the tutorial first, then attempts to solve the problem independently. The other (Approach B) is best suited for the student who prefers to devise his own solutions first and, only after, validates his approach by checking the tutorial registration and other online resources.
Students can use the tool to two distinct advantages. First, they have a visual representation of the various paths they can take to complete the module and assess the progress they have achieved so far as shown in Figure 2 (The stars represent successfully completed steps, the grey circles are current activities and the empty circles represent future steps).

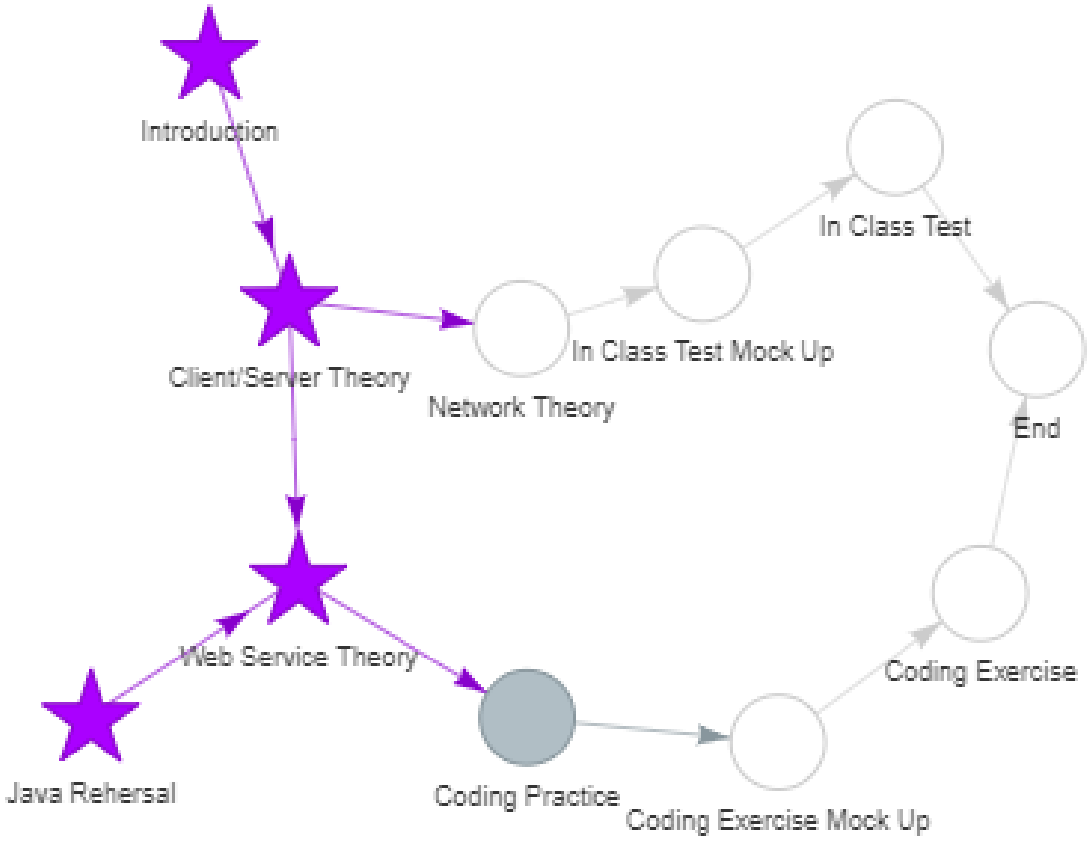

Figure 2, Student View of the Module Progress

Furthermore, as students' progress through the module, they can also track whether their choices were successful or not, thus ending up in creating a map of their own personal inclination and strategies that will help them in taking better choices In the future. As an example, in Figure 3, the personal choice record shows a high success rate for the red approach (attending guided tutorial first) and a less successful outcome for the blue approach (trying to find solutions before attending the guided tutorial).

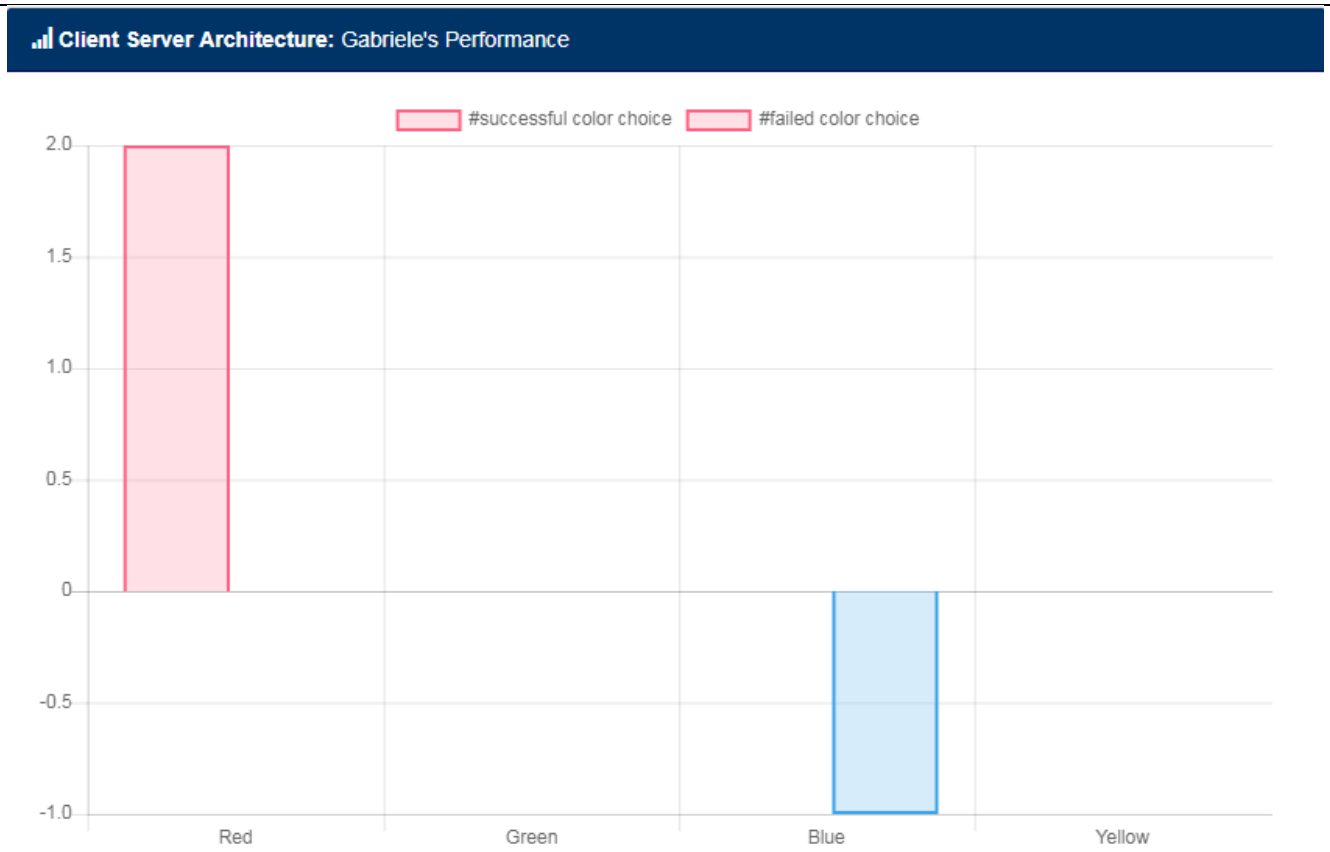

Figure 3, Personal Choice History

Students can decide to experiment various approaches to see which one fits them best but they can also take a separate but conceptually related psychometric test call Lumina SPARK (https://luminalearning.com/) that will provide them with an estimate of their profile.
In 2017-2018 we ran a pilot scheme for the students of the Mathematics for Computing Module and In 2018-2019 we are going to engage a larger number of students in the same level 4 Module of the School of CS & Engineering, and also students of the level 5 module Client-Server Architecture.
We believe that our developments could be taken further by colleagues across UoW and we are currently preparing a news story about EnAbled which was invited by the Centre Teaching Innovations. We are particularly interested in supporting early levels of the university education by helping students during their transitional period.
EnAbled was shown at the EC-TEL conference in Leeds (https://my.ltb.io/www/#/) and published in the related proceedings (http://ceur-ws.org/Vol-2193/paper15.pdf). We have also presented the prototype at another, national event (EduTech Show at Kensington Olympia - October 2018 - https://www.edutechshow.co.uk/home), where we have observed great interest from the teachers and head teachers attending the show; this has suggested that students in the pre-university education could be another possible target audience.
EnAbled is also used as part of an experimental setup between UoW and WIUT where we are exploring its usability across different cultural backgrounds. The on-line version of the EnAbled prototype is available at https://uowenabled.herokuapp.com/

Aiming at developing more EnAbled functionalities to meet students and academics needs, our project, STRIVE, has been based on EnAbled.

the following two research questions for STRIVE were set:

1.) How the Learning Compass software, recently developed within Quintin Hogg Project “Enabled”, can be further enhanced to assist first year students in selecting optimal learning strategies?
2.) How the Leaning Compass” can be further enhanced to help students in the assessment phase?

The two research questions set above determined the main aims of the project:

1.) What features can and should be added to the "Learning Compass" application to make it more helpful to the first-year students in their journey through the liminal space – the transitional period in which many new comers to the university find themselves.
2.) In the current version, "Compass" does not address and is not linked to any of the formal assessment patterns. Hence, the second aim of the project is for the project team to investigate how the assessment components can be incorporated into the "Learning Compass" application.

To achieve these aims, the following objectives have been set:

1. To carry on the research on possible feasible improvements of EnAbled including the module representation and the interface.
2. To conduct an analysis of the activities related to the assessment for a dedicated module and of the functionalities needed to effectively learn dedicated topics to define areas of intervention:

Tips:

1. *the aim is the 'what' of the research, and the objective is the 'how'. You can list your objectives with bullet points or numbers.*
2. *When working out who the 'stake holders' are, think about who you want to respond to your findings and carry out some changes.*

Section 3. Methods (150-300 words)

To achieve the objectives above the following methodology was followed:

The research questions mentioned in Section 1 suggested that during the project team initially had to study relevant literature and work on specifying the requirements that will address both research questions. This stage involved

- Literature research – learning styles, user interface, modern solutions
- Running the software with students to gain their feedback

At the same time the student team was familiarised with "Compass" architecture and its implementation.

The second stage was based on the following methodology:

- Research on relevant technological solutions to design and implemented the user requirements
- Implementation of the solutions found.

To ensure the knowledge transfer from David, who in particular developed EnAbled, a series of meetings dedicated to this issues was organised. This was particularly important as our student team faced a rather difficul task – to work with the code that has been already developed. The whole setup was new to our students, includir the MongoDB and the graph based implementation of the module representation in EnAbled. It is known that in software development it is often easier to write the code than to learn the code written by somebody else. Otherwise, the team has followed the software development methodology – from gathering the requirements to th implementation and testing through the modelling. These also form part of our results and will be described in Section 4.

Tip: It would be useful to attach a copy of the Participant Information Sheet and questionnaire you used, as an Appendix at the end of the report.

Section 4. Results

In this section we describe two main streams of work within the project.
The first is improving the EnAbled functionality by linking the modules.
The second is work on enhancing EnAbled with the assessment.

Concerning the first stream, the team worked on the possibility to implement a system which allows related/ pre-requisite modules to be linked together, as well as depicting it using the vis graphic application.
With little to no previous experience or knowledge of the programming languages and API utilized in the project, students had to do brief tutorials and exercises on the following: NodeJS, Express, MongoDB, BootStrap, Javascript, JQuery and vis.
As a result, the interface of a dependent dynamic dropdown on the module edit page was developed. This would allow one to pick a module, then a topic of choice to link to the selected module. However, the technical difficulty in finding relevant programming solutions was too high. In particular, it was found that one of the most difficult issues was saving the links through MongoDB. The team has also researched the ways of using vis, but it was found that most of the times it would overwrite and display the nodes of the relating module but not the parent nodes, and other times it would not show the linked relationship. These problems became open and would need more time or a more experienced programmers to deal with.

The screenshots below in Figures 4-6 illustrate the team adventure in solving these problems.

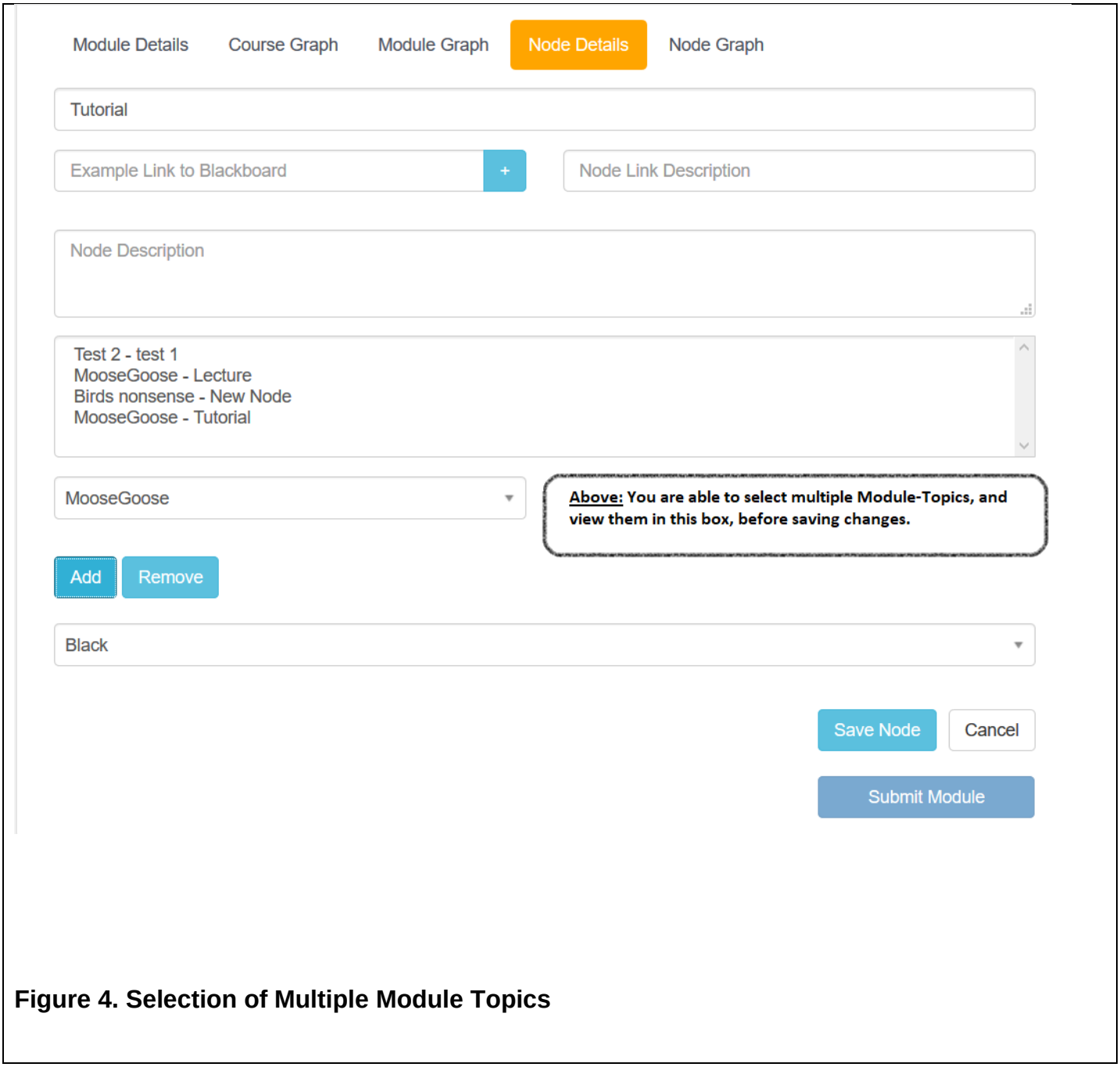

Figure 4. Selection of Multiple Module Topics

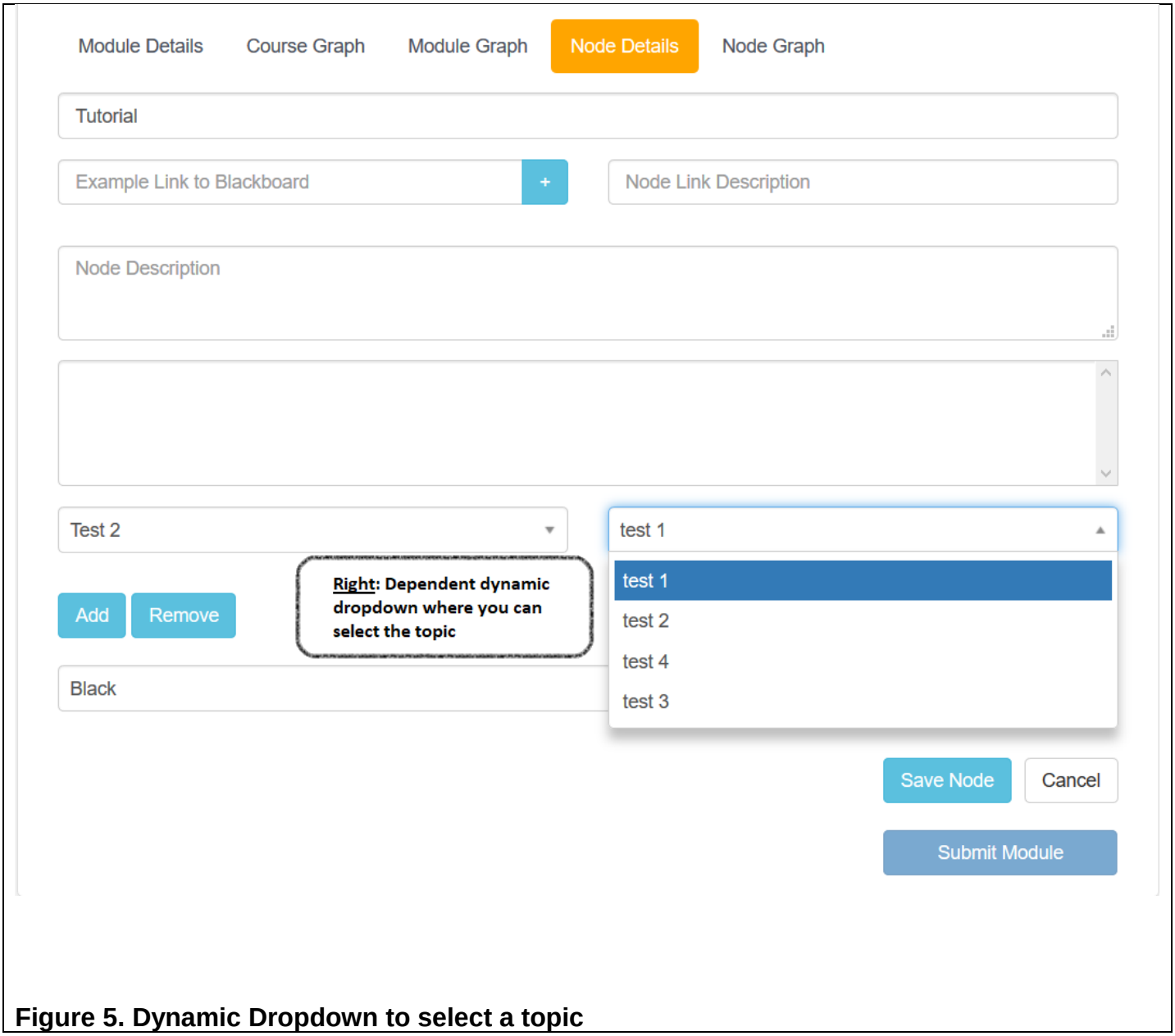

Figure 5. Dynamic Dropdown to select a topic

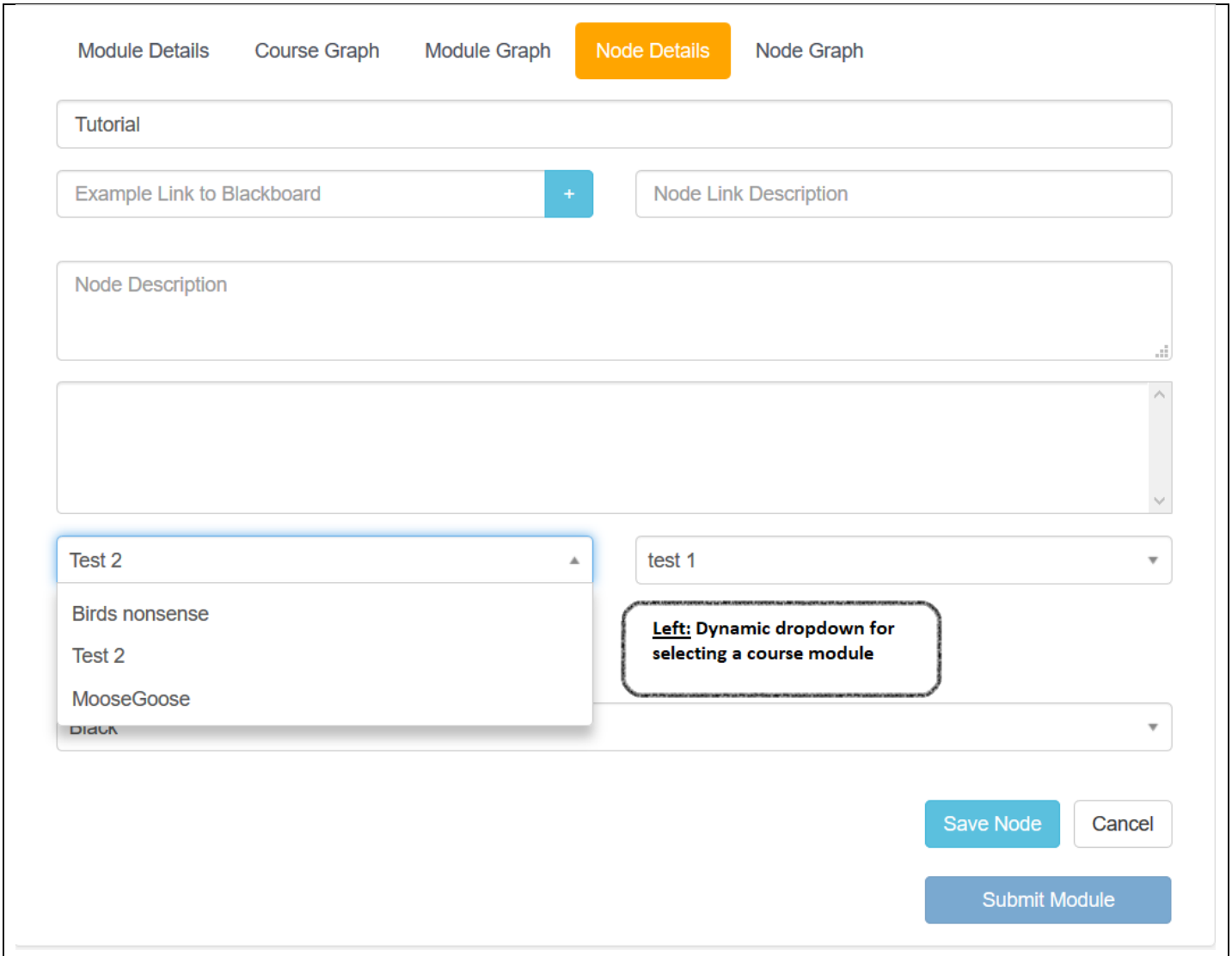

Figure 6. Dynamic dropdown for selecting a module

Concerning the link between EnAbled and the assessments, first, a deep and detailed analysis has been conducted on how the EnAbled platform could be best used for a specific module (5COSC004W Client Server Architecture)
In particular, the analysis of the activities of this module and of the functionalities needed to effectively learn the topic, was carried out. This led to the precise definition of three areas of intervention:

2) Although EnAbled has a workable interface that points to all the teaching material on Blackboard its usability could be greatly improved, and it not was particularly attractive to students.
3) Apart from the poor usability of EnAbled GUI, another fundamental flaw was spotted. There was nothing specific to the module in either EnAbled or BlackBoard. One could use EnAbled to navigate the material on BlackBoard and to make decisions among different Learning Paths but there was nothing specific neither in BlackBoard or in EnAbled to that particular module.
4) Two fundamental functionalities of great usefulness to both students and members of the module team were formulated.
    a. An interactive environment to learn how to:
        i. Write a Client from an Existing Server
        ii. Write both a Client and a Server from specifications
    b. An interactive environment to test the code implemented in the point (a)

c. An automated grading system that supports the members team in grading large numbers of exams that consist in code developed in point (a). Each student had to produce code to cover both (i) and (ii) and the grading work was significant given that the number of students was around 200. That meant 400 exams (i and ii), each consisting of several files. That means grading thousands of files.

All these have enabled the design and implementation of a complementary software, called ClieSer.

Interactions among ClieSer, EnAbled and BlackBoard/Panopto

The overall interaction among **ClieSer, Blackboard/Panopto and EnAbled** can be recapitulated in ***Figure*** 7 below.

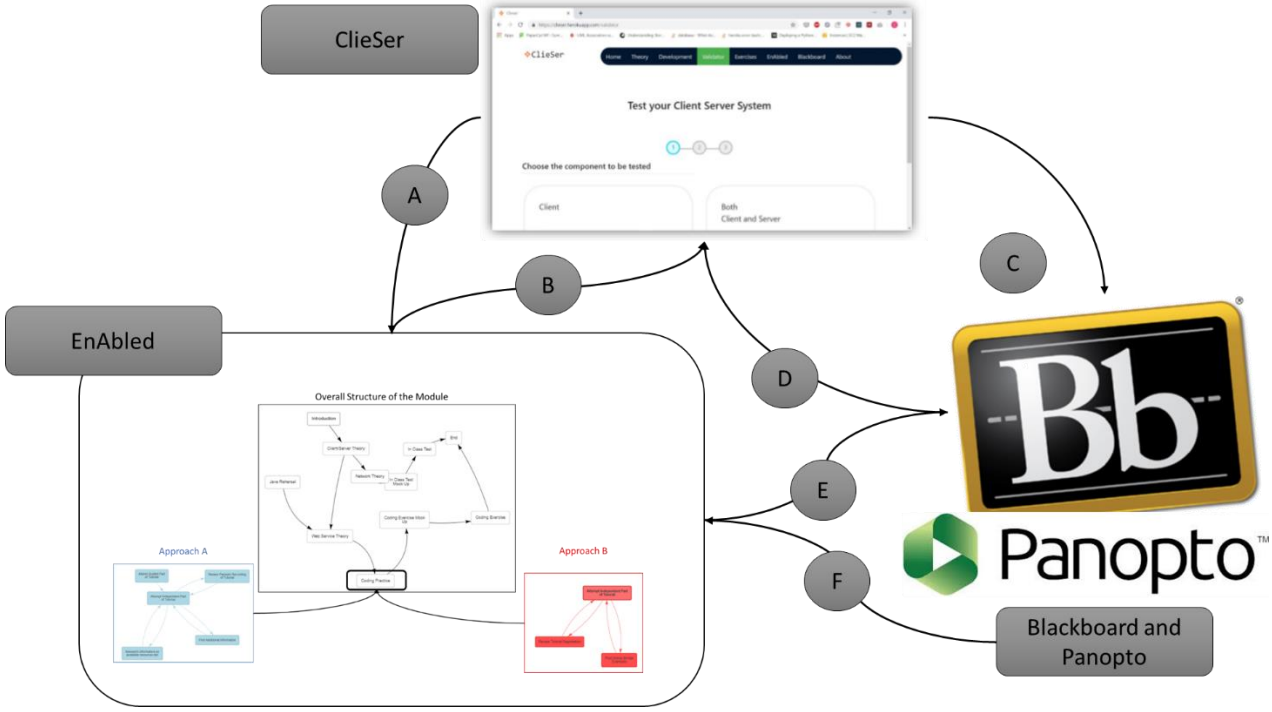

Figure 7. Interactions among ClieSer, EnAbled and BlackBoard

Students can login in any of the three system and navigate from one to the other. Unfortunately, the Authentication of Blackboard and EnAbled are not connected yet so that, one first time, students must login separately in each system. Students can use the EnAbled interface to analyse what is their progress and which learning activity they want to engage in. At this point, they can:

- use the Blackboard/Panopto material (link E),
- if they find the material of the material on Blackboard/Panopto unsatisfying or incomplete, navigate from BlackBoard/Panopto to ClieSer and use its additional material (link D)
- go directly to ClieSer (Link B)

On the other end, students can start by logging into blackboard and:

- Work directly in Blackboard/Panopto, a more traditional approach,
- Move to EnAbled to see what is their progress and what Learning Activities they have to do next (Link F)
- Move to ClieSer to use its advanced learning tools (Link D)

Finally, Students can login directly into ClieSer, and:

- Study in ClieSer
- Move to BlackBoard/Panopto to find additional material (Link C)
- Move to EnAbled to what is their progress and what Learning Activities they have to do next (Link A)

Main functionalities of ClieSer

The main functionalities of ClieSer are listed in its home page.

- Theory: offers a set of web pages with information on Client/Server theory, a closer interaction via links to Panopto and Blackboard is still needed here.
- Development: helps the student to learn how to develop a Client/Server with step by step concrete examples
- Exercises: offers the students exercises and automatically validates the code being developed (it uses the same engine that is used for the automatic grading of the exams).

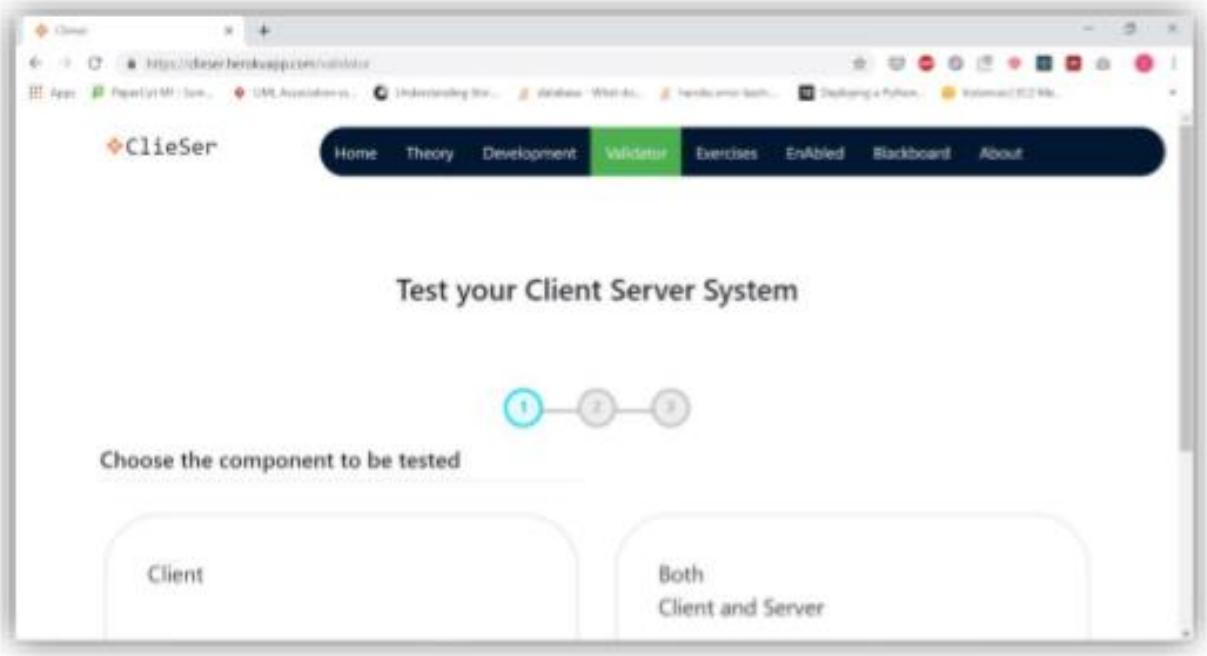

Figure 8. Main Functionalities of ClieSer

Architecture of ClieSer

The architecture of ClieSer is worthy of mention on its own, it has been carefully designed with abstraction, flexibility and reusability in mind. Its main features are that:

1) Nothing in the core architectural structure is linked to the specific module so that it is easy to adapt to other similar module
2) Nothing in the code of evaluation engine (to evaluate the student exercises and the automated exam grading) is specifically tied to the Client/Server paradigm. It relies on matching two pattern of messages (the specified pattern and the submitted one) between two digital entities. So that, instead of checking that the Client and Server exchange the right type of messages, I could check instead that a program is sending the correct messages to a file, to a database or to another program.

This design covers all aspects of the program from the Architectural Stack to the Class Structure, to a standardized REST interface to the architectural stack to the deployment system based on Heroku (Figure 48 to 10)

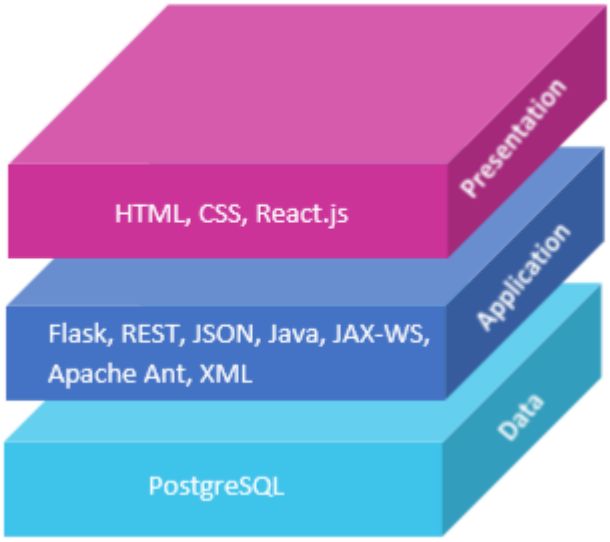

Figure 4, Architecture Stack

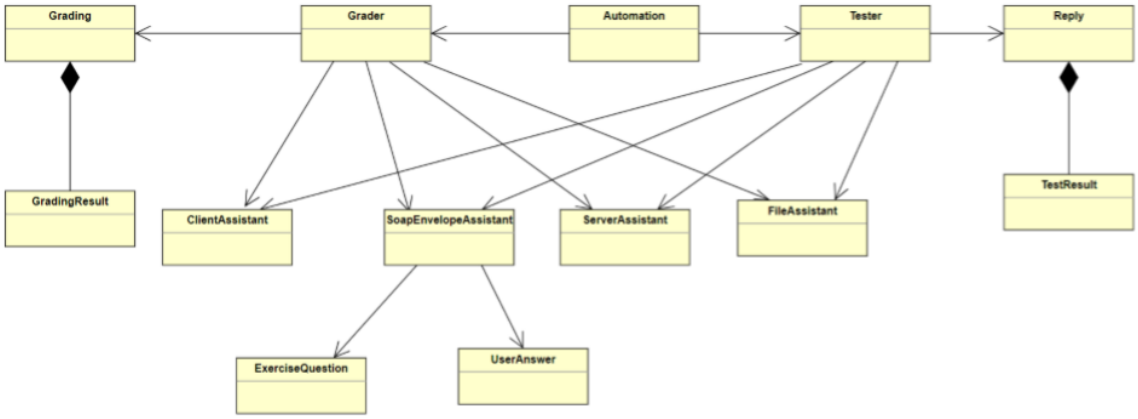

Figure 9. Overview of the core code structure - High Level Class Diagram

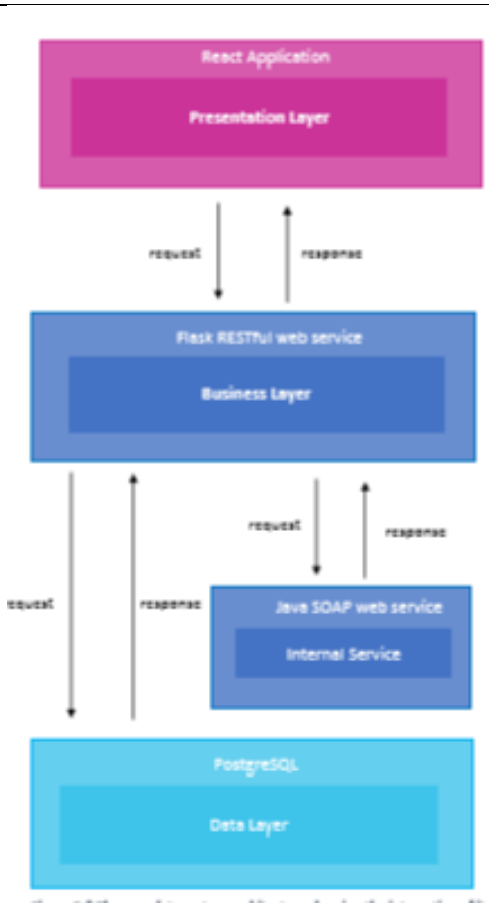

Figure 10, Complete System Architecture

Tip: When writing this section, look at your graphs and tables and ask yourself – what does this data mean?

Section 5. Discussion (300-600 words)

Our choice of the methods to achieve the aims was determined by various factors. First, the student team should have gained sufficient knowledge on the existing software, its architecture and specifics of the implementation. Subsequently, the work followed the software development research methodology. Additionally, as of the importa tasks was to grasp the existing and functioning software, the reverse engineering approach, widely used in this area, was explored.

The work on the software development was guided by the software development methodology. Much of it was related to the findings of the ways to add the functionality to the existing software which gave the team ideas of th functional requirements of the desired software to meet. Once the requirements have been identified the design stage involved modelling together with the choice of programming solutions. This mainly related to the work on linking EnAbled with the assessments. For the programming solutions of this scale, it was necessary to come up with good understanding of the software architecture which was another part of the software development methodology. The compete system architecture was given in Section 4. A good piece of knowledge for the students involved into project was to carry on what Is called proof of concept of the architecture. In other words, a pieces of the system design should have been tested to be able to “talk to each other”.

Finally, we gained feedback on EnAbled from the students of the foundation year at our partner institution – WIUT – when the software was offered to them for the revision on Mathematics module in November 2018. This additionally, informed our work, with the clear message to make the interface more attractive and appealing.

Tip: When writing this section think about your aims. Did you achieve your aims? Were your results as expected? Are you findings consistent with other work in the literature?

Section 6. Conclusions and Recommendations (200-300 words)

The project has been successful in many respects but has also revealed issues to take on board for the future similar projects. STRIVE offers significant research on the issues of EnAbled functionality, its link to the assessment and the interface. We have also managed to test the software with the students of our partner institution – WIUT – while we were offering students in Tashkent to use EnAbled during their revision for one of th modules in the winter session in November 2018. The latter has also revealed an obvious way to improve the software – make it attractive to the students. This was outside the scope of the project but represents an importa task for the future.

The adventure of the team dealing with the project setup was quite educational. We believe the following issues should be taken into account for any future similar SCC project that involves software development.

1. The scale of our project appeared to be very large and level 5 students found this quite challenging while level 6 students, who themselves are involved in their final year project (often of a similar scale) are more comfortable here
2. The project by its formulation suggested to work on improving the existing software – EnAbled. Learning the technicalities of the code written by other people, is often a challenge and it was a challenge for our student team.

To make the work within the project like STRIVE more efficient, we believe that the student team SHOULD includ student the equal number of students from levels 5 and 6. Ideally, the topic of the final projects should be relevant – in this case, level 6 students would be able to serve as SCRUM masters and will be able to lead the software development and chare their expertise with level 5 students.

Tip: Keep your stakeholders in mind when writing this – this section needs to be specific for your audience (who will pick up on your recommendations?). You can list your recommendations with bullet points but you should remember to include a statement to conclude your report.

Section 7. Dissemination (200-300 words)

The project's results have been already partly disseminated among the colleagues within the School of CS& Engineering. Our trial of the software at the WIUT has given another boost to the project and as part of the projec with WIUT we are preparing a journal submission on the development of threshold concepts in mathematics and reflection in EnAbled. We will again use EnAbled during the 2019-2020 academic year and will disseminate the outcomes of the project among our colleagues to encourage its use in teaching. Similarly, we will make sure that the outputs of the project will become familiar to the students. Subsequently, the dissemination of the project results is thought to be productive among the academics and student of the department of CS&Engineering. EnAbled has been also demonstrated at the Educational show at Kensington Olympia in the autumn 2018. This has proven that EnAbled is also useful for the use in secondary education. However, any success here would depend on the enriching the software functionality and improving the interface. For the latter, funds will be neede and therefore any dissemination targeting potential sponsors will be very useful – educational funds, research councils, trusts.

Tip: Think about how you are going to get your research across to your stakeholders. Be realistic about this an consider whose help you may need in the process. You will also get help from the Students as Co Creators team with the dissemination of your project.

Section 8. Reflection (200-300 words)
As we mentioned several times this project has been rather challenging for all participants – both academics and students. The main challenge was the scale of the problems to solve. These, overall, were much higher than even the standard final year project. Subsequently, the student force was distributed according to the weight of th problems. Our final year participant was working on linking EnAbled with the assessment which level 5 students were looking at the possible improvements of the system, selecting those to be targeted. Even this task was quite challenging for level 5 students. As mentioned above, the project assumed work on improving the existing softwa hence learning the code written by other people. Although this is what students often do applying reverse engineering, the success here depends on the scale of the software and in our case, it has been a challenge. To make the work within the project like STRIVE more efficient, we believe that the student team SHOULD includ student the equal number of students from levels 5 and 6. Ideally, the topic of the final projects should be relevant – in this case, level 6 students would be able to serve as SCRUM masters and will be able to lead the software development and share their expertise with level 5 students. *Evaluate*: What was good/effective about the research experience? What problems did you have? *Analyse*: Why did you have those problems? *Plan*: What would you do to overcome the problems? What would you do differently next time?
Tip: Think about what lessons you have learned and conclude on your project.

Submission Instructions:

Please ensure the academic partner has checked a draft report and attach the cover sheet at the front of your report. A final report should be emailed to studentpartnership@westminster.ac.uk by the 24th July 2019.