Software Engineering Group Report CMP020N204S

Submitted to The University of Roehampton

By: David Kahsay, Linda Osman, Sadia Ali, Samira Mohamed, Summer Link

Declaration

We hereby certify that this report constitutes our own work, that where the language of others is used, quotation marks so indicate, and that appropriate credit is given where we have used the language, ideas, expressions, or writings of others.

We declare that this report describes the original work that has not been previously presented for the award of any other degree of any other institution.

Table of Contents

1. Introduction

In this project, we aim to create a website where a variety of reports based on population information will be able to be accessed via different webpages linked together from an SQL database. This includes a variety of variables that we would break down from the SQL database to create the different reports including these from the back to front end.

For our problem regarding this project, we have based it off user stories and personas in which will be the base that we will use for our website use and permissions. For this, we have based it off a school system, where there is a teacher and a student who equally want to access the website for these reports. Whilst the student will access the website to study for their GCSEs, the teacher will access the website to create lessons from the analytical information that we are giving out via these reports.

We aim to have at least 3 fully functioning reports on the website that are linked together by hyperlinks/buttons from the navigation bar. To achieve this, we needed to work together as a team in an agile scrum way to tackle our problem in small pieces. This allowed for us to have regular meetings regarding delegating tasks as well as meetings where we would be more productive, discussing our tasks and ensuring that they would go out on time. Using the Kanban board we create, we could outline the tasks that needed doing and the status on how they were going. We would also regularly upload updated work to the GitHub to make sure that we were all using up-to-date work from each other's tasks.

First, we would break up the sprint into equal parts where each person would have a task to work towards to achieve a result of the full sprint being completed. To do this, a meeting will be conducted where we would break up the tasks into where we would most feel comfortable.

The kanban board would then be used to note down what tasks are being carried out, by who, when they will be needed and the current progress on it. This way we were able to keep track of where we were always on the project.

Issues are to be to be noted on the GitHub so we know where problems may lay in our team.

Then we would go and do our task, meanwhile scheduling another meeting in place to ensure that each person will be able to bring something to it regarding their task, or we would be able to help where needed.

Once any task is finished, it would be updated on the kanban board and uploaded to GitHub to ensure that we share what we have done with the team.

The cycle would repeat itself until all our tasks have been completed to the best of the teams' abilities.

For legal issues, a big one facing the website would be the login details which each user would be using. The biggest part of this would be the password that the user created to login, concerns about where this is stored and protected would be a big issue for user safety. So, to ensure that this would be less of a concern, we would have to hash each password and username that is used to log into the website within our database.

Where we use the SQL database, we should be sure to use correct credentials for the rightful owner as we do not want to use any copyrighted material without permissions from the original owner. This would prevent any legal issues regarding the database we have used.

For group working, we should ensure tasks are delegated equally to make every person so a fair portion of work to prevent any unfair overworking from individuals regarding this project.

A code of conduct is to be created and signed by the team to ensure that each member knows what rules the team should follow, and sanctions will be given out if incorrectly followed.

This project overall is well suited to our module as our main aim is to work effectively as a team, using different tools as measures to ensure this is done correctly and efficiently as a part of a team. Everyone taking the same user stories and personas into account allows for us to work around a centre piece of the vision we want to achieve. Having a variety of people doing different tasks has let us open to the idea of project management from different perspectives as we would all micromanage our own work as well as our teams to make everything line up to what it needs to be. Using professional tools such as GitHub and kanban has also allowed for the team to be more organised out of their comfort zone as everyone needed to make sure they were doing their part and consistently taking care of each platform to communicate with the other team members.

2. Literature or Technology Review

In this project, we used a range of technologies to create our application.

For this project, we could have used Pug or HTML for the front end of this application. Our web browsers only understand HTML and it is the most widely used in web development. Pug templates must be converted into HTML. [1]

For the front end, we decided to use Pug. Pug is a template engine in node.js Pug structure is quite easy to read and is syntax sensitive and it returns HTML. [2] The res. render function in Pug is done to make this conversion.

For the backend, we used node.js and express.js. Node.js is a server-side run time that runs JavaScript not in a browser. JavaScript makes server-side applications. Express.js connects the front and back end. In our index.js express.js is used. We create routes for all the pages. Without this, the server will not receive the requests for the pages. It allows the content of our Pug pages to be displayed.

We used a Docker file to run our applications in a container through docker-compose. This also connects the back end and front end.

For our agile project management, we decided to use zube.io. Zube is a collaboration tool that we chose to use as it can link with GitHub. Zube has a clean layout and is extremely easy to use and navigate which is important as this is where most of the organization takes place. In our Zube we had a kanban board and a sprint board. This is where we kept track of our product backlog, who was assigned which role and what each sprint entailed. This was key in how efficiently our team could communicate and keep organised.

For testing, we used Circle.ci. Circle.ci is a continuous integration tool. We also had the option of using Travis.ci. Cicle.ci supports a range of programming languages and does not require any installation. We were able to set up testing as we could access Circle.ci through GitHub. Cicle.ci is easy to use and completes testing quickly.

To make our wireframe we used Figma. Creating our wireframe on Figma allowed us to have created a visual representation of our website that we could reference while completing our project. This helped us tremendously throughout the development of our application. Figma is easy to use and flexible and allows multiple people to collaborate. We chose Figma over options such as Mockflow, lucid chart.

3. Design or Methodology

For the project a team has been designated and we worked by building our application in sprints. The ongoing process of continuation of our application has made a remarkable progress at the end.

It was potentially a systematic utilization of engineering principles in the creation and advancement of software.

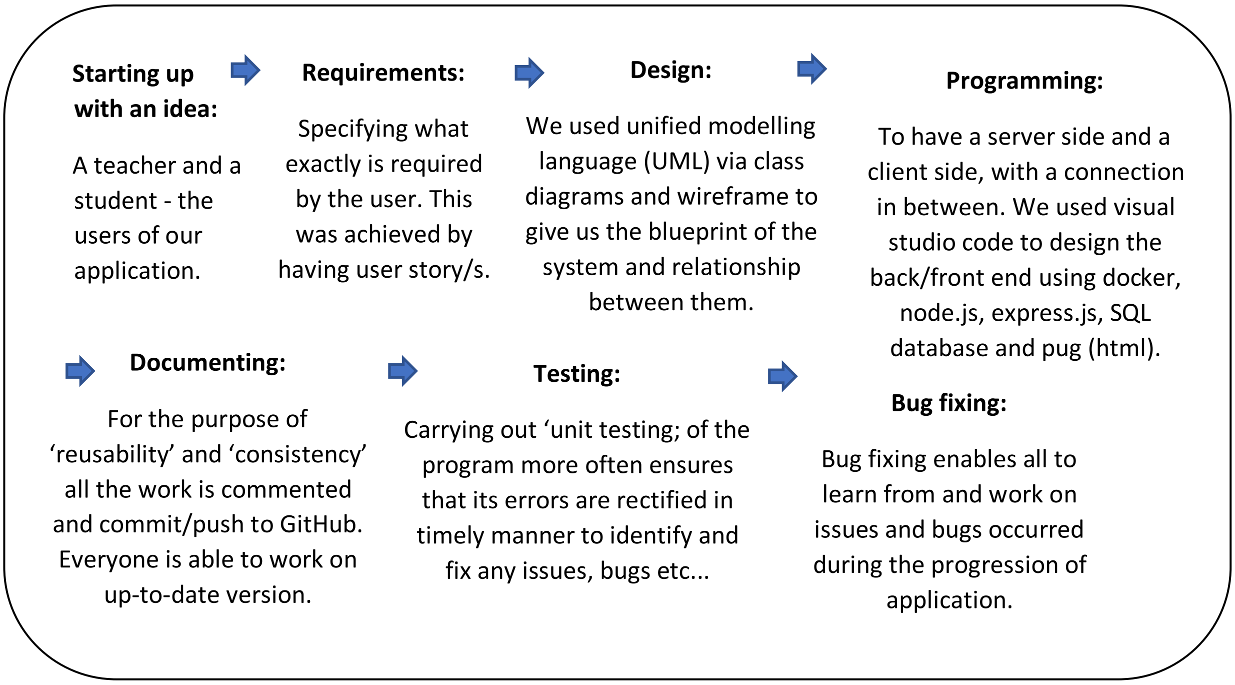
How are we going to undertake the project?

Working in a series of step is known as methodology and according to the requirements, Agile method was the best option. As it focuses more on ‘human-centric’ ideas although can extend the iterative ideas too.

Another advantage of Agile method is that the software is progressed or evolved by the collaboration between developers and client.

The process is supported by multiple parts through self-organizing teams and as the requirements and challenges change, teams can be Agile and adapt accordingly.

Process involved in developing a software (from moodle - week one):



Starting up with an idea:

A teacher and a student - the users of our application.

Requirements:

Specifying what exactly is required by the user. This was achieved by having user story/s.

Design:

We used unified modelling language (UML) via class diagrams and wireframe to give us the blueprint of the system and relationship between them.

Programming:

To have a server side and a client side, with a connection in between. We used visual studio code to design the back/front end using docker, node.js, express.js, SQL database and pug (html).

Documenting:

For the purpose of 'reusability' and 'consistency' all the work is commented and commit/push to GitHub. Everyone is able to work on up-to-date version.

Testing:

Carrying out 'unit testing; of the program more often ensures that its errors are rectified in timely manner to identify and fix any issues, bugs etc...

Bug fixing:

Bug fixing enables all to learn from and work on issues and bugs occurred during the progression of application.

I summarised the process in above figure.

Software Design:

Firstly, a code of conduct was set in place for all team members to work in harmony and to achieve targeted goals.

Then we had a product backlog implemented in Kanban board, which guides the team members in what they need to achieve and what their working goals are.

For the kanban board we used 'zube.io', through kanban all team members can collaborate, and everyone is acknowledged of what assigned task/s needs doing or in progress, what assigned task/s is already done, what needs altering and what the escalated issues are and how to resolve them.

All team members joined platform 'GitHub' so they can push their work and everyone including teacher can see the collaborative work.

'Dockerfile' was set up to help communicate server side (backend) and the client side (frontend) by containerisation.

User stories using unified modelling language (UML)

By defining use case in full, suitable 'use case diagram' and a 'class diagram' to aid in software design.

Having wireframe in place to see how the application flows together and for that we used the collaborative interface design tool called 'Figma'.

For the backend we worked on JavaScript (node.js and express.js). As for the frontend we worked on Html (pug and CSS). SQL for the database and queries and 'Dockerfile' for the front & backend. This was all done in Visual Studio Code.

Time to time we carried out 'unit testing' and 'integration testing' to ensure that our application is taking shape bit by bit in the right direction. Otherwise, it would be unwise to wait till the end and the application crashes or filled with bugs/errors. Timely testing helps revise and fix things to avoid waste of time and hard work. To carry out tests we used 'Circle.CI'.

Alternative Approaches

There are so many software and tools for different approaches and requirements.

Similarly, we had so many options to choose from when building our app.

For the kanban board instead of zube.io we could have opted for: Trello, Jira, GitHub project etc...

For the UML diagram instead of draw.io we could have opted for: lucidchart.com.

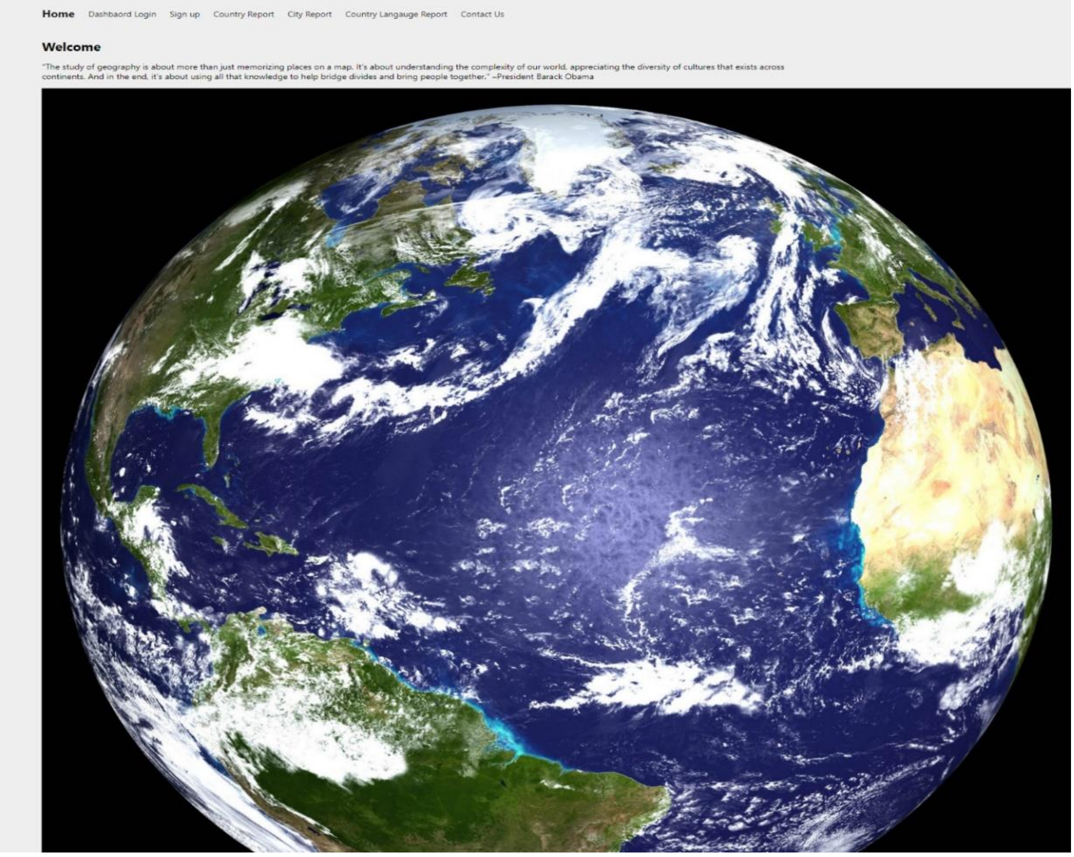
For the wireframe instead of figma we could have opted for: mockflow, lucidchart, ninjamock etc…

For the unit testing which supports CI (continuous integration) instead of circle.ci we could have opted for: Travis.ci, CruiseControl[3], Hudson[3] etc…

4. Implementation or Results:

The outcome of the collective project is a website that is connected from the backend to the frontend. The backend had the data of the world SQL. It was designed in a way to gather all the data to the different countries and populations without any problems, allowing all access to the frontend. The database was implemented with the pre-filled data provided which granted us time to focus on the node.js file, keeping in mind that we split the group to collaborate on each section of the program. The frontend of the website was able to clearly display certain pieces of data depending on what button we clicked on, and this was due to the PUG and CSS files created to interact with the node.js.

The results of the website were entirely constructed around the wireframe. In our results, there was a homepage that exhibited titles for the country report, city report and country language report; it also had an image of the world signifying the context for the webpage and increase visual appeal for user engagement. We also implemented a sign up as well as a sign in page for students and teachers/administrators which had different permissions and accessibilities based on how you signed in. If you the user signed in as a student, then they could have access to quizzes and to the country, city and language report which came with information of population and other relevant data. However, if you signed in as an administrator then you would have special permissions to create quizzes for the students in addition to hiding and showing it. The administrator also had access to various reports similarly to the student's access. A unique feature we did apply was a search button so that the user can comfortably find data relevant to what they are looking for and if the user inputs a country that is not within the data it will output “country not found” as we had also added an error handling mechanism called try and catch. There is also a contact us page for any queries the user may want to ask.



Welcome

"The study of geography is about more than just memorizing places on a map. It's about understanding the complexity of our world, appreciating the diversity of cultures that exists across continents. And in the end, it's about using all that knowledge to help bridge divides and bring people together." –President Barack Obama

Figure 2

The website may have fulfilled its purpose, but it still does have its differences with strengths and weaknesses. The strengths include a user-friendly interface, interactive quizzes and the incorporation of error handling to lessen frustration from the user. Although, we had many strengths the program still had its minor setbacks for example one of objectives of the start of the project was to implement a way for the admin to change or add to the existing data. The code implementation had been inserted but the program wasn’t successful in that aspect. Nevertheless, there is still ways to access that command now as the group have worked around this difficulty.

Time limitations were an issue as there are still many unique ideas to be built onto the website to improve design appeal or inclusivity for users who may have disabilities. Nonetheless the final product we created in the time given is more than satisfactory with the objectives we issued from the beginning.

5. Conclusion

We worked in a scrum team to develop a crud application in this group project. A CRUD application is when we use the front-end to a database, providing a technique to help create, read, update, and delete data from a website to a database on a server.

In this group project we frequently used tools that is common for many software development teams uses such as (visual studio, docker, and git).

In our project we should have:

- Our application must be version controlled using Git and publicly accessible on GitHub. So, the teacher can access it
- Our application has to use a database for storage.
- Our web front-end must use PUG
- Our web back- end must run via Node.js
- Lastly, our application must be deployable as a Docker container

Reflection

Our group we have learned many things and one of the main things we learned creating the front-end and the back-end of the website and we connect both through express.js. This is our first time creating a website through HTML. Even though we are proud as a group to make an amazing website with many features such as a login page for both students and teachers d how they both have unique features on the website such as the teacher can assign reports and grade students and t and the students cannot do that. However, as a group, if we had more time to complete the coursework, we should have taken me with the design aspect such as make the home page more visually pleasing by adding more smaller pictures instead of one big picture and just have more fun with designing it in general.

We worked effectively as a group, as we created a WhatsApp group chat so we can easily contact each if we got problems or if we want to do a team meeting. This made group organised as we all know what to do. However, the main issue our group had is making sure we have a specific time to make a group call as some of the group members have jobs or there were other factors such as the Wi-Fi not working or family issues. To combat this issue, we should of meet up after our lessons at university so every Monday, Wednesday, and Thursday we all would be free to discuss the work and other outside factors such as Wi-Fi wouldn't be an issue.

Some people in the group had poor management skills such as we found it hard to balance this group project with our other work in the other modules. This because we must meet up as a group and join calls so we would be spending more time in this module then the others. To overcome this issue, we would make a timetable to dedicate our time to each module and free time so we could equally spread our time with other modules, so we will not fall behind.

Some of the group members felt as if two people building the website was too much so as a group, we should have split up the work better as the two people felt as if they had too much work and responsibility. However, as a group we overcame this issue as one of the group members mentioned the issue in the group chat and we have gotten more help and felt as if the burden was released from their shoulders.

One of our best assets as a group is encouraging each other in the group chat we will motivate each other to get the work done so we are more likely to do the work in higher standards as we do not want to let the group down.

Lastly, as a group, we can criticise each other work to make sure we are handing in the best work. For example, Samra would send the class diagram and use case diagram to the group chat or to the members individually to get different ideas so she can improve the work.

Future Work

In the future our group could focus more on the design aspect of the coursework so that will be our focus next time.

We could have made the website more user friendly by changing the colour theme of the website so it can be more eye catching to the users. We always must make sure the colour contrast is high enough so people visual impairments would use our website. Additionally make sure we have headings and subheadings so its organised so the user can easily navigate around the website.

6. References

[1] *Learn About Pug In Node.js*. Available: https://www.c-sharpcorner.com/article/pug-in-node-js/.(Jul

[2] *Pug Vs HTML*. Available: https://www.c-sharpcorner.com/article/pug-vs-html/.(Jan 21,).

[3] “Agile Testing - Tools,” *www.tutorialspoint.com*. https://www.tutorialspoint.com/agile_testing/agile_testing_tools.htm (accessed Apr. 27, 2023).