**Module: Synoptic Project**

**Assignment: EWB Digital Design Challenge**

**Set by:**  Aaron Bostrom a.bostrom@uea.ac.uk

**Date set:** 30/05/2022

**Value:** 100%

**Date due:** 16/06/2022

**Returned by:**

**Submission:**  Blackboard & Presentation

**Learning outcomes**

* Analyse the social and cultural contexts of the Engineers without borders brief to design and create a computing based prototype.
* Communicate effectively in a variety of professional contexts.
* Apply software development fundamentals to create and test a computing based prototype.
* Function effectively as a member of a team engaged in activities appropriate to software development.

**Specification**

**Overview**

You and your team are required to design, implement, and test a software solution to solve the Engineers without borders (EWB) challenge. This software prototype will be designed to solve a local problem or aid the local community. The solution will be contextually aware of the local requirements and the issues and problems they face.

**Description**

You will design and implement a software prototype as a group of 5 students. You are required to produce a supporting documentation, a working prototype and then present your solution at the end of the project.

**Relationship to formative assessment**

The synoptic project is designed to utilise and demonstrate the application of the skills you have developed throughout your studies. This is an opportunity to put your practice into action, by applying these skills to a real-world live brief. This means that there will **not** be any additional taught material to that already presented thus far in your degree. If you need help or support, we can provide it, but you should also refer back to previous content where applicable.

**Deliverables**

Working in your group you are required to produce three types of deliverables.

**Design**

Using your knowledge of systems development methodologies and documentation you need to create a **presentation** outlining the software system you have designed. This **must** include UML diagrams, or if your prototype depends on more than one piece of software you could include layer or stack diagrams, along with requirements gathering, design decisions and user testing.

Your presentation will include discussion and explanation of the tools and libraries you chose to produce your solution.

Your presentation will include analysis of the problem domain, and research to evidence and defend the decisions you have made with respect to the brief.

**Implementation of Prototype**

Based on your design you will implement a software prototype to the specification you have laid out. This software prototype can take whatever form is necessary to solve the area of the brief you have identified. It is recommended you effectively utilise the skills of your chosen group.

Considerations you should make are, does your solution scale well, does your software require resources exceeding the capabilities of the local users, does it have prohibitive computing costs? Does it require advanced technical knowledge to maintain? Does it require internet access?

**Software testing**

Software testing is an important aspect of ensuring your software works correctly. If your software forms part of critical infrastructure software failing or breaking can have disastrous consequences so ensuring it functions correctly is imperative.

You should include some form of user testing. User testing is performed on users **external** to the project.

**Presentation and Group Working**

At the end of the project, you will be expected to give a 12-minute group presentation explaining your project and how it answers the brief. This presentation will include a short, pre-recorded demonstration of your prototype. You will have 3 minutes for questions from the panel of two assessors.

In addition to your group presentation, you will be expected to attend a weekly scrum meeting with a member of the teaching team. During this meeting you will demonstrate and evidence how you worked as a group. Dividing up tasks, co-ordinating work, and managing workloads. You will be expected to use tools such as Trello and Git to support sharing your code and managing the different tasks that need solving for your project. These meetings will be mandatory and will contribute to your final mark.

**Resources**

* <https://stackoverflow.com/> - An excellent site for finding information about specific issues relating to various programming languages, including Python. It is important however not to become too reliant on sites such as **StackOverflow**, which are great for details, but don't give the ``big picture'', so it is difficult to get a good understanding of programming in this way. Note that if you re-use or modify code found on-line, then you must provide a comment giving the URL and a brief explanation of what modifications have been made. Re-using code found on-line without proper attribution would constitute plagiarism.

If you become stuck, please do feel free to discuss problems with the teaching

assistants during scheduled laboratory hours, or contact me via email ([a.bostrom@uea.ac.uk](mailto:a.bostrom@uea.ac.uk), dave and debbie) for assistance outside lab hours.

**Marking scheme:**

|  |  |
| --- | --- |
| **1st Marker Name** | **Student Number** |
| **Date:** |  |
| **2nd Marker Name:** | **Date:** |

|  |  |  |
| --- | --- | --- |
| **Marking Details** | **Mark %** | **Marking Comments** |
| Design: Requirements  UML Diagrams, Agile methodology, Stakeholders, Persona, Empathy map, MoSCoW, User stories etc... | 20% |  |
| Design: UML Diagrams | 10% |  |
| Implementation: Sustainability  Region appropriate software development and deployment | 15% |  |
| Implementation: Maintainability  Code commenting, naming conventions and software structure | 10% |  |
| Implementation: Functionality  Implementation of design. Usability. Modularity, and well-defined interfaces. | 20% |  |
| Testing  Presented evidence of User testing | 5% |  |
| Quality of Presentation  Group presentation outlining what your group has made. | 10% |  |
| Teamwork and Attendance  Evidence of teamwork. Trello tasks. Attendance of each weeklcaly scrum meeting with a member of teaching team | 10% |  |

Extra Comments:

Total Score