

**ARLA – Alumni Registration and Linking Application**

**B.Sc. (Hons) in Software Development**

**Student: Ciaran Roche**

**Project Partner: Pavel Antonov**

**Supervisor: Damien Costello**

**Submission Date:**

**Table of Contents**

1. Introduction
2. Methodology
3. Technology Review
4. System Design
5. System Evaluation
6. Conclusion
7. References
8. Appendices

**Introduction**

The project that was undertaken is called “Alumni Registration and Linking Application” also known as Arla. What is Arla? It is an application that was originally intended to be developed on behalf of the college itself. The original idea was that GMIT was to use it on their website. Arla is where users can sign up and login to the application using their google email address. They will then be able to register their details such as their name, the course that they studied in GMIT and the year in which they studied it, their interests and where they are now. The users of the application, once logged in, can then view the connections to a course on the graph page. On this page it will have every user that has linked to said course. Users will then be able to click on a person they wish to connect to. When doing this a popup will appear and a button which will bring them to a messenger page of the user they wish to contact. From there the users can now message each other.

The application is designed so that it is very user friendly to people of all ages and all experiences of using technology. For example, it was intended to make sure that someone who has never used a computer before in their life has as much ease of using the application as someone who uses it every day. The user will also have as much flexibility as they like when it comes to the information that people are able to access about them on the website. The user may decide they don’t want to have any connections and not upload any information to the application. Perhaps they just want to have a browse on the application without actually connecting to people.

It is a good project to develop at level 8 as there is so many different ways the project can be implemented with, given enough time, many cool features can also be implemented. Another reason as to why it was a good project for a final year project is because of the complexity and workload of it. These technologies used were extremely complex and took a lot of research and reading of documentation to fully understand and implement.

The front end of the application was developed using React. The graphical part of the application was implemented using D3.js. There were some new and exciting libraries that were implemented into the application also. These were chatengine.io and react popup as well as some libraries that were intended for use but could not properly use for one reason or another. These technologies and libraries will be discussed some more in the technology review section of the document. The application is fully responsive on the front end that can only be accessed if the user has logged in. It is also set up on Heroku so that it can be accessed publicly. Below at the end of the introduction are links to the application hosted on Heroku along with the GitHub repositories where the work was stored.

The backend had many issues in the development process which will be discussed more in the methodology section and outline the course of action that my supervisors advised to take. The backend connected to the new technology Neo4J which is a graph-based database where all the information passed from the front end was to be stored. The graph on the React front end was essentially the Neo4J database superimposed using D3.

There were many goals set at the beginning of the project before development initially took place, and it was hoped that as many as possible if not all of them were implemented and completed by the end of the project life cycle. The goals that were set were as follows:

The first goal of the project was to create a fully responsive web application that is hosted on a middleware site such as Heroku or AWS that allows for users to connect to each other and message each other via a messenger styled page on the application.

The second goal was to accommodate all people who wish to use their website, make it extremely simple, efficient and user friendly for both a user with no previous experience of using technology and someone who has high levels of experience in using technology.

The next goal in developing the project was to allow the user to enter as much or as little detail as they wish and to decide how much of this information would be visible to other users of the application. Along with this, it was also a goal to make sure that the user can adjust or remove any details they may have inadvertently added to their profile.

Another goal that was set for the project was to create a way in which users can create their own groups and contact each other. Perhaps along with this create their own little mini graph and show all the people within the group. They could then have a group chat on the messenger page for example.

The next goal that was planned was to gain a greater understanding of the technologies that were going to be used in development and to learn some new and exciting technologies such as D3 and Neo4J in the process.

Along with learning new Technologies there were some new things that have not been thought on the course before and it was exciting and intriguing to see can it be done for this project. These were: to create a google login application and understand the mechanics of what is behind this and understand how to implement this for future projects. Another of these is to have a dynamic home page for the user and understanding how to implement this. In other words, when the user logs in, I want it to show only their individual details on their home page, so that it would be different for every user of the application. The next goal was to understand how to draw graphs and essentially understand the basic concepts that were needed for this application using D3. It was also preferable to improve any testing skills throughout the duration and at the completion of programming the application.

As well as software development goals, there were some goals that involved project management. The project cycle was ideal to improve any ability to use the Jira application and learn how it can be incorporated it into the project. This is so that it improves efficiency when planning and implementing a project during a project life cycle. Other applications that were ideal to incorporate fully and improve usage of were OneNote for documentation and GitHub for the programming aspect of the project.

Finally, in terms of goals that were set out for the project, it was ideal and preferable to improve soft skills. Improving any skills such as teamwork and communication by having regular meetings with the supervisor and teammate. This is ideal to constantly keep people up to date with the project as well as making sure it is on the right track.

Methodology:

The methodology chapter in this paper is where the projects methodology will be discussed and essentially how the project was carried out and implemented. The development approach that was used will be discussed under topics such as, how it faired and an evaluation of the style that was adopted. As well as this the planning stages of the project will be discussed. An in-depth look will be taken at what technologies that were used in the early days to set a plan out for the project, how this developed as time progressed and how changes were dealt with. A thorough and in depth look at the technologies that were used for documentation, planning and designing of the project. The tools that were used to develop the project will be discussed and analysed, for example, GitHub. The research that was conducted in the planning stages of the project will be thoroughly explained, dissected and analysed. Also, a delve into the weekly meetings and communication throughout the project will take place.

Technology Review:

The technology review section is where an in-depth look will take place at all the different technologies that were used in the development process of the project. Some of these technologies will include react, D3, chatengine.io and all the different libraries that were either implemented successfully or implemented unsuccessfully for one reason or another. A brief look will be taken as to why said technologies were not implemented and what alternatives can be or were used instead. Each technology will be analysed to the fullest extent.

System Design:

The system design section is the part in which a detailed explanation of the overall system architecture will be given. This is essentially the HOW of the project. It is where the knowledge gained from research is implemented. Also, it will look at each aspect of the application and give a detailed overview and in-depth analysis of different components of the system and how they work together.

System Evaluation:

The system evaluation section is where an evaluation of the system is conducted. For example, what is good, what is bad, what needs more work and what could be done to make this even better. An evaluation of goals will be made, where they met, if not why?

Finally, a conclusion to the project will be given and an overall opinion of how the project went. Analysis will take place to see what can be done to improve the project as well as performances of members. An evaluation will be made on what was learned during the project cycle. Below are links to the front-end GitHub repository, the backend GitHub repository and the link to the Heroku hosted application.

[**https://github.com/CiaranRoche203/Arla-App-FrontEnd**](https://github.com/CiaranRoche203/Arla-App-FrontEnd)

[**https://github.com/CiaranRoche203/Arla-App-Backend**](https://github.com/CiaranRoche203/Arla-App-Backend)

[**https://arl-application.herokuapp.com/**](https://arl-application.herokuapp.com/)

**Methodology**

There was a lot of planning, research and decision making that went into the early days approach to the project. The first and most important aspect was to set out a stable plan that the team members can follow. The first step in planning that took place before all else was to research heavily and thoroughly. Research is a common practise that took place in the duration of the project.

The first stage of research involved which type of methodology suited the project and the team members best. There was a general idea of which methodology suited the project best before any research was done but it was important that this was done correctly as it would benefit the planning and overall flow of the project. The 2 cycles that were researched were the Waterfall and Agile.

After much deliberation between team members and with advice from the supervisor, it was decided that an Agile approach benefitted the project the most. The reason for this is due to Agiles’ flexibility. It also was a preferred methodology to follow as it allowed for consistent testing of the application as well as receiving consistent feedback. Waterfall could and probably would have caused a lot of issues overall if it were the methodology that was pursued. The reasoning for not taking on the waterfall model is that it is too rigid. In order to move on to the next step, i.e., design to implementation, the design phase must be one hundred per cent complete.

It was then decided that Agile would be the development methodology to be followed. Agile allowed for the project to be developed while minimizing risk when adding new functionality. Some of the risks that it minimized where bugs and changing of requirements. The consistency of testing the application that would take place with Agile would cause less issues in the long run and would not delay development as much as waterfall potentially could. The flexibility was a key factor for our choice of Agile. The ability to make continuous changes and additions to the application coupled with the fact that these new additions and changes would be tested consistently immediately after implementation was a major factor in the decision.

The next stage in the development of the project was to plan. The plan is crucial as it sets a basis of what needs to be done, by when and by who. The plan is a fundamental necessity to this or any project as without one, it will be a disaster so to speak. The first stage of planning was to hold a meeting with the supervisor. In this meeting the overall project was discussed with the objectives and tasks being set out from the beginning. An idea was given as to what the final project should eventually look like. This was used as a guideline, but changes could always be made to add cool new features to the project. The next stage of planning involved designing the system. It was imperative to get an understanding of what would be happening with the application, what sort of data would be passed from front to back end, where it will be stored etc. A simple architecture of the solution was drawn up (see system design). Following this, some front-end design was drawn up to get a basic understanding of what a final product may look like (see system design diagrams also). Discussion between team members and supervisor was had over the potential design and some changes and additions were made accordingly. The final stage of the first phase of planning was to set up a Jira board. On this Jira the following were set up:

* Roadmap

Graphical user interface

Description automatically generated

The roadmap is a useful tool that allows for a visual representation of the project progress. It is essentially a Gannt chart. The sprints could be adjusted accordingly as needed. For example, if sprint one was taking longer than initially planned, simply moving the bar on the chart allowed for the sprint to continue until the new date it was moved to.

* Backlog

Graphical user interface, text, application

Description automatically generated

The backlog was where any tasks that were incomplete in the sprint were held. It was easy to access and see what is holding the project progress up and who has been assigned the task.

* Board

Graphical user interface, application, Teams

Description automatically generated

The board is a cool feature that basically has 3 categories as seen above. It categorises what needs to be done, what is in progress and what has been completed. Once all tasks have been completed the sprint can be completed and the next sprint will begin.

Tasks were separated into sprints. There was an idea to have three sprints. Getting each sprint done and moving on to the next sprint was ideal, however there was also flexibility. Should a task or feature take longer than expected, it was agreeable that sprints could be adjusted accordingly, or the task could be moved into the new sprint.

Sprint 1:

Sprint 2:

Sprint 3:

Planning – drew out some design, discussion was had and changes were made between the team. Set out documenting the planning stage using one note and jira. Used a gannt chart to set out a project timeline. Assigned tasks to each member, changes could be made to the gannt chart as time went on. Made a plan for most important features to implement and had a plan if things went bad what could be sacrificed. Mention sprints and what was in each one.

Meetings were held every week with both team members and supervisors.

Sprints

We continuously tested the app as we made it. Testing for every aspect of what could possibly happen and testing any change no matter how minor. Time consuming but worth it for the steady progression of the project.

GitHub was used as a tool for storing and collaborating on work mention each repo. One not for documentation.

Mention the research taken. What was researched etc give some references to the research.

Issues problems and how they were solved.

Approach to the problem

[**https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/**](https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/)

**Technology Review**

**System Design**

**System Evaluation**

**Conclusion**

**References**

**Appendices**