Global Classroom

Project Plan and Risk Strategy

E – Voting System

DT282/3

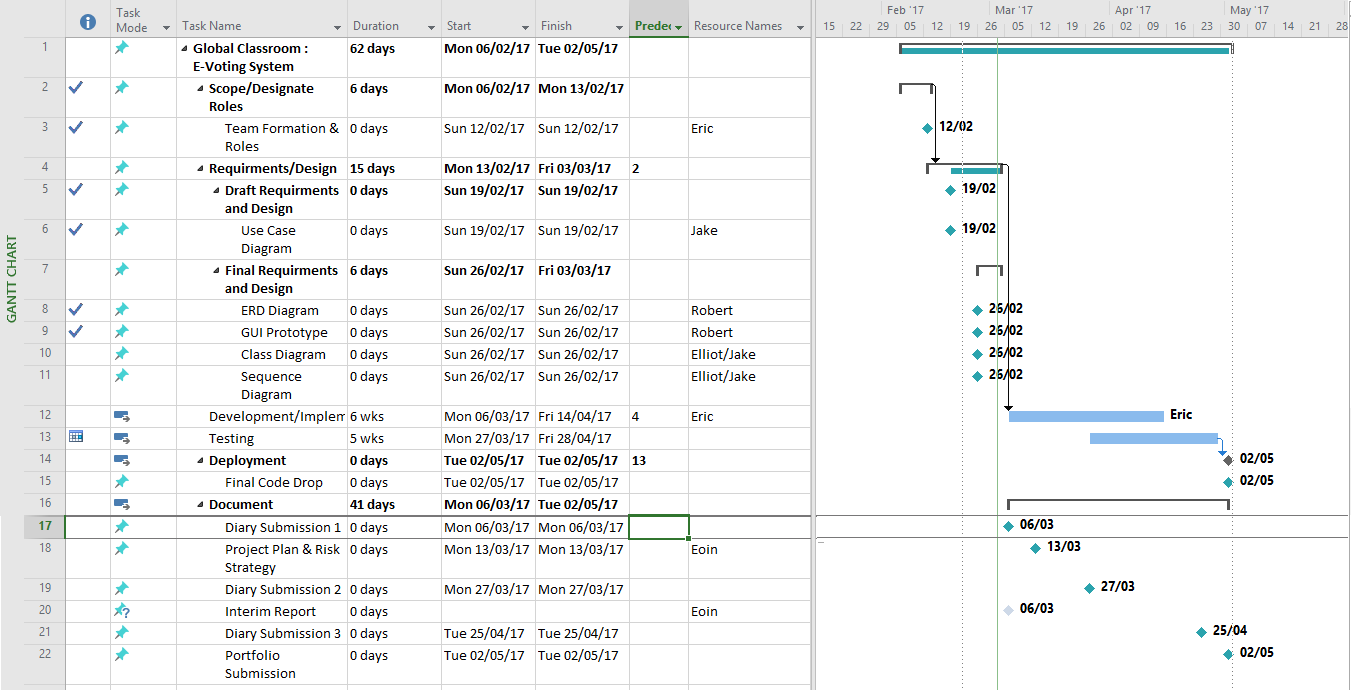


Student Names & Numbers:

* Eoin Smyth [C14427202@mydit.ie](mailto:C14427202@mydit.ie)
* Robert Wijntjes [C14356786@mydit.ie](mailto:C14356786@mydit.ie)
* Jake Young [C14706715@mydit.ie](mailto:C14706715@mydit.ie)
* Elliot Igoe [C14733411@mydit.ie](mailto:C14733411@mydit.ie)
* Eryk Szlachetka [C14386641@mydit.ie](mailto:C14386641@mydit.ie)

Due Date: 17/03/17

# Project Plan



The full MS Project file will also be included in the submission, along with a rough initial Development Plan.

# Risk Strategy

## Communication

* A major risk that needs to be addressed on a project like this is one of Communication. The nature of this project unfortunately makes communication a potential problem in every aspect. We are operating in three different time zones at any given time, which can lead to confusion and otherwise easily avoidable mistakes when something is not communicated clearly. A direct example of the impact of the different time zones is the limited window available for us to have face to face meetings due to the 8-hour difference.
* Of course, not all problems with communication stem from the international nature of the course. It is easy to unknowingly miscommunicate without meaning to, and if not addressed early it can be become problematic later.
* **Solution**: Communicate clearly with fellow team members. Allow for time difference, and be understanding of potential miscommunication.

## REST Architecture

* Rest Architecture is the architecture we chose for this project as it has many benefits that suit our design of the E-Voting system, such as being based on a Client-Server architectural style and by being stateless. However, that does not mean that there are no risks associated with this architecture.
* The downside of it being stateless is that there may be latency in request processing times and bandwidth usage, which could impact on the user’s interaction with the application. However, this risk is minimal if considered during the application’s design.
* REST architecture utilises HTTP, which means any risks associated with HTTP must be considered when using this architecture. The main issue with HTTP is security, which of course needs to be addressed due to the nature of the app. It is paramount that the user’s personal information, and any sensitive information the user inputs, is protected. While, in theory, ISP’s and local governments having access to this information shouldn’t be an issue, things like proxy servers and DNS spoofing should be considered.
* **Solution:** Focusing heavily on security is a main priority. An option that could be considered that will address much of the security concerns would be by adopting HTTPS over HTTP.

## Design Phase

* Much of the risks associated with the Design Phase are tricky to identify, as the problems they cause may not become apparent until further down the line in the Development Phase. As such, is very important that there is little ambiguity in our designs, and that as a group we are happy to continue to the Development Phase when we deem appropriate.
* **Solution:** Minimise ambiguity in design.

## Development Phase

* A big problem that we could face during development is unfamiliarity with the environments and technologies in which we are developing. We are all comfortable developing in Android, HTML and PHP, for the most part, but have far less experience developing in Node.js and Heroku, which we will also be using. Unfortunately, due to time constraints, we don’t have the ability to extensively research and prepare to develop in these environments. This presents with us the need to learn as we develop, which has its own risks and problems.
* While this does present us with an opportunity to learn a new set of skills, it can hinder the development time of individual aspects of the project. Something that might only take a half hour to someone with experience developing in these environments may take double or triple the development time.
* **Solution:** Ensure accuracy during development. While it may seem initially longer, it is easier to prevent a problem from happening rather than fix it.