*Ryan Goudie*

*Group 1 project plan*

Ryan Goudie, Cian Burns, David Maxwell, Daniel Taylor

Table of Contents

[**Group Members:** 2](#_Toc64462669)

[**Overview** 2](#_Toc64462670)

[**Requirements** 2](#_Toc64462671)

[**Objectives** 2](#_Toc64462672)

[**General Approach** 3](#_Toc64462673)

[**Gantt Chart** 4](#_Toc64462674)

[**Pert Chart** 5](#_Toc64462675)

[**Task Breakdown Table** 6](#_Toc64462676)

[**Work Breakdown** 7](#_Toc64462677)

[**Roles** 8](#_Toc64462678)

[**Major Deliverables** 8](#_Toc64462679)

[**Evaluation Methods** 9](#_Toc64462680)

[**Resources Required** 9](#_Toc64462681)

[**Contractual Agreement** 9](#_Toc64462682)

**Group Members:** Ryan Goudie, Cian Burns, David  Maxwell, Daniel Taylor

# **Overview**

We will be creating a Football Tournament Program based upon 20/21 Euros League. The aim is to create a system with a given list of teams and randomly generate groups of 4 upon which the top 2 performing teams will proceed to the knockout stage which the team that wins across 1 leg will then proceed. Major milestone for the project will be the implementation of each class as will be detailed in the Schedule. In terms of constraints, we are limited to 24 teams from the commencement of the tournament at the group stage.

# **Graphical user interface, text, application, email Description automatically generatedRequirements**

# **Objectives**

* We will develop a terminal based java program
* We will be using a database back end
* We will be creating a user-friendly experience with clear readability and ease of access
* After development the user will have a full system to create and simulate the Euros Tournaments

# **General Approach**

We are going to be using the Rapid Development Cycle

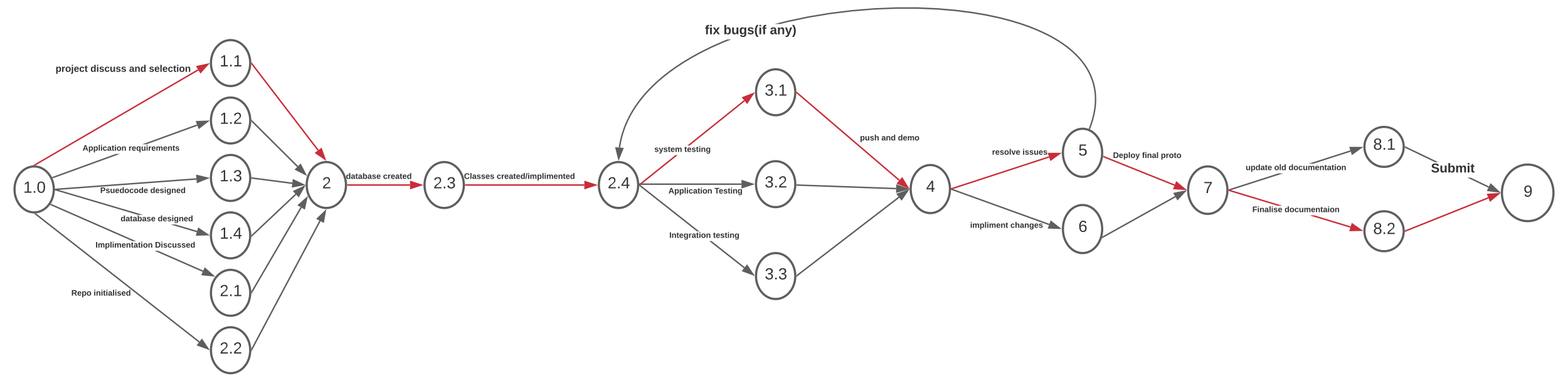
We will be using Java to implement this project with use of randomising algorithms. A CSV document will be used to include the teams into the program. We will be using GitHub for version control and documentation.

# **Chart, bar chart Description automatically generated with medium confidenceGantt Chart**

**Chart, funnel chart

Description automatically generated**

# **Pert Chart**

****

# **Task Breakdown Table**

|  |  |  |
| --- | --- | --- |
| **WBS number** | **Task Descriptor** | |
| **1.0** | **Project discussed and topic selected** | |
| **1.1** | **Applications Requirements** | |
| **1.2** | **Pseudocode Designed** | |
| **1.3** | **Database Designed** | |
| **2.0** | | **Development** |
| **2.1** | | **Implementation Discussed** |
| **2.2** | | **Repository Initialised** |
| **2.3** | | **Database Created** |
| **2.4** | | **Classes created and implemented** |
| **3.0** | | **Testing** |
| **3.1** | | **System Testing** |
| **3.2** | | **Application Testing** |
| **3.3** | **Integration Testing** | |
| **4.0** | **Push First Prototype and demonstrate** | |
| **5.0** | **Update and resolve any issues** | |
| **6.0** | **Implementing said changes** | |
| **7.0** | **Push and deploy final prototype** | |
| **8.0** | **Documentation** | |
| **8.1** | **Update Pre-existing documentation** | |
| **8.2** | **Finalise final documents / talk materials** | |
| **9.0** | **Submit** | |
|  |  | |

# **Work Breakdown**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Work Breakdown Structure** | | | | | | |
| **Deliverables: Euro football tournament program**   * **1st Prototype** * **2nd finalised prototype with updates** | | | | | | |
| **Start date:**  Week 1 | | **Due date:**  Week 12 | | | **Project duration:**  12 weeks | |
|  | | | | | | |
| **Task** | **Immediate Predecessor Tasks** | | **Estimated Time Duration** | **Estimated Resources** | | **Assigned To** |
| Project Planning | Project Discussed | | 1 Week | Word Processor, Github, Discord | | Everyone |
| Development | CSV File Created | | 3 Weeks | IntelliJ ,Github, Discord, Project Database | | Everyone |
| Testing | Randomiser Class Created | | 2 Weeks | IntelliJ ,Github, Discord, Project Database | | Cian |
| Push Fire Prototype and Deploy | Integration Testing | | 1 Week | IntelliJ ,Github, Discord, Project Database | | David |
| Update and Resolve any Issues | Integration Testing 2nd Run | | 1 Week | IntelliJ ,Github, Discord, Project Database | | David |
| Implementing Said Changes | First Prototype | | 1 Week | IntelliJ ,Github, Discord, Project Database | | Everyone |
| Push and Deploy Final Prototype | Updates and Fixes | | 1 Week | IntelliJ ,Github, Discord, Project Database | | David |
| Documentation | Finalised Prototype | | 1 Week | Word Processor, Github, Discord | | Ryan |
| Submit | Finalised Report | | 1 Week | Word Processor, Github, Discord | | Everyone |

# **Roles**

Ryan – Development – Documentation

Cian – Development - Testing

Daniel – Development – Database

David – Development – Planning / Deployment

# **Major Deliverables**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Major Deliverable** | **Ryan** | **Cian** | **David** | **Daniel** |
| **Project Plan** | **R** | **C** | **C** | **C** |
| **Implementation plan** | **S/A** | **R** | **R** | **R** |
| **Database design** | **S/A** | **C** | **S/A** | **I** |
| **Project Development** | **S/R** | **R** | **R** | **R** |
| **Testing** | **S/A** | **R** | **C** | **C** |

# **Evaluation Methods**

**System Testing -** A system test is used to ensure all parts of the app will work when run together. For our Tournament app, this means that our different classes and functions (e.g. Randomiser or Teams) work cohesively and produce the expected results.

**Unit Testing –** A unit test will be used when we complete 1 segment or class of code, this will be used to ensure that this section of development works as intended and doesn’t house any bugs within.

**Application Testing -** Application test involves running a unique test scenario on the app to see if the expected results are produced. We will carry out manual testing of specific use case scenarios to test the boundaries of the app.

**Integration Testing -** Similar to a system test, integration tests check to make sure all separate parts of the app can work with each other as a pair. This means making sure that the Teams class can read the CSV, or that the randomiser can produce points that create the group stage leaderships.

# **Resources Required**

Access to the list of teams that are in the tournament.

Access to Github

Access to Project Database

Access to Discord

Access to IntelliJ

Access to individual hardware – 3 Windows Desktops

1 MacOS laptop

# **Contractual Agreement**

We aim to have the project done by 26th April 2021 and for it to be up to standard and have the correct deliverables