**IFAF Rules Change Proposal and Voting System**

**System Requirement Specification**

[up746804]

[up746804@myport.ac.uk](mailto:up746804@myport.ac.uk)

*School of Computing, University of Portsmouth, Buckingham Building, Lion Terrace, PO1 3HE*

1. Introduction

This document concerns the *rule change proposals and voting* system proposed for IFAF (International Federation of American Football). The system is to be used by all registered members of the federation as well as by the system administrators. As such, access privileges with different level of functionality and usability need are to be integrated. The purpose of this document is to give an insight into the scope, functionality, usability, security and integration requirement of the proposed system in all level of access.

1. Aim

The IFAF members are invited to propose a change in IFAF rules each year. Currently, there is no software system to capture the proposed changes. In addition, voting on the newly proposed change is done either verbally or via e-mail. A new system is proposed that will digitalize the process of proposing a change, voting on the change and approving or declining the proposed change. The aim of the project is to design, build and test the proposed system to enable members to propose a change, view others proposal and vote on the proposal. It is also to enable administrators to control user accounts, accept or refuse new users, approve or refuse new proposals, remove any proposal/comment the administrators deem unnecessary or inappropriate.

1. Key personnel

Jim Briggs – System owner

Aseged Woldeselassie – System Design Authority

1. Rationale for development

Currently, there is no system as such used for proposing a change of rules and voting on the proposed change. This unavailability of convenient system might lead to a reduced number of participants in both proposal and voting process. The proposed system will greatly improve members participation and transparency of voting. It will also be integrated with existing website for ease of use.

1. Interactions and dependencies

The system will be developed to run independently of any existing system, apart from existing IFAF website. All users, existing and new must register with the new system before they are able to use it. However, if there is an existing login system with username and password, data migration could be considered. This might include migrating all existing rules and codes used to identify/tag rules on which change is being proposed.

1. Timetable and Scope

This project is scoped to be standalone. As such, the owner has the choice of running the system as part of existing website or use different domain. In terms of system development, the scope of the project is to enable users to propose changes to existing rules (tag to existing rule) in the form of text. It is also to enable users to comment and vote on proposed changes by others. Additional functionalities for administrators such as ability to control activities on user account is within the scope of the project. Search functionalities will also be added for use by both users and administrators.

The following table shows general deliverable time

|  |  |  |
| --- | --- | --- |
|  | Deliverables | Delivery date |
| 1 | System specification requirements | 01/02/2018 |
| 2 | User interface and code structure design | 15/02/2018 |
| 3 | First iteration (Alpha) | 08/03/2018 |
| 4 | Second iteration (Test) | 20/03/2018 |
| 5 | Second iteration (Beta) | 26/03/2018 |
| 6 | Beta testing (demo) | 29/03/2018 |

1. Data requirements

The following data dictionary indicates system critical data;

7.1 Internally sourced data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attribute | Required | Type | Field length | Default Value | Validation | Comment |
| User ID | Yes | Int. | Incr. | None |  | Users are given incremental user ID. Only used by the system |
| First Name | Yes |  |  |  |  |  |
| Last Name | Yes |  |  |  |  |  |
| Username | Yes | Text | 50 Char | None |  | On registration |
| Password | Yes | Text | 50 Char | None | Text/number comb. | On registration |
| E-mail | Yes | Email |  | None | Email |  |
| Country | Yes | Look-up |  |  |  |  |
| Role/s within the system | Yes |  |  | None |  | Role given to the user, not required from the user. |
| Roel (IFAF) | No |  |  | None |  | Required from the user on registration. |
| Proposal | No | Long Text |  | None | Not empty |  |
| Comments | No | Long Text |  | None | Not empty |  |
| Vote | No | Look-Up |  | Abstain |  | For, against, abstain |
| Tag | No | Look-Up |  | None |  | Tag current rule code/ID |

7.2 Externally sourced data

The data requirement below is optional and could be migrated from existing website/database.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attribute | Required | Type | Field length | Default Value | Validation | Comment |
| Rule | Yes | Text | 50 char | None | None |  |
| code/ID | Yes | Text | 50 char | None | None |  |
| Alteration code | No | Text | 2 char | None | None | Denotes (Mutual consent, management only) |

1. Functional requirements

The following functionality requirements have been identified;

8.1 Users

* Enable users to registration/login
* Enable users to propose a change or rule/s in the form of text
* Enable users to amend or remove a proposed change (rules apply)
* Enable users to comment on a proposed change
* Enable users to amend or remove comments
* Enable users to vote on a proposed change
* Enable users to change or remove vote on proposed change (rules apply)
* Enable search functionality for searching proposals

8.2 Administrators

* Enable administrators to approve/decline new users
* Enable administrators to approve/decline proposed change
* Enable administrators to remove/hide proposed changes or comments
* Enable control over user accounts/activities including changing roles and suspending users.
* Integrate system with existing website (optional)
* Two step registration must be implemented for administrators to approve users.
* Show result of votes on proposed changes (optional)
* Enable search functionality for searching proposals

1. non-functional requirements

9.1 Users

* Filtering functionality for proposals (popular, most recent, or specific date)

9.2 Administrators

* Enable analytical report/ close changes and comments on proposed change (optional)
* Enable email (SMPT) functionalities for notification
* Username/password retrieval functionalities (optional)

1. User Interface

10.1 Users

User interface design is available in the next deliverable.

Building a user profile where the user can manage a personal profile is optional but not a requirement. Hence, user will see proposed changes (index page) when first signing in. From there, all functionalities will be available for the user. Each proposal (sorted in ascending order) will show its corresponding comments, vote numbers and status.

10.2 Administrators

Administrator interface design is available in the next deliverable.

The administrator interface will be the same as the user interface with some additional control enabled. These functionalities will enable the administrator to control user account actability.

1. Integration

The system will be designed to be standalone from initiation. However, the federation has existing website. Integration of the proposed system into the existing website is plausible. Further, the federation might wish to perform data migration so that current users are automatically enrolled on the proposed system. The exact integration requirement is yet to be established, thus not a functional requirement at this point.

1. Development hardware/software

No requirement has been specified in terms of development platform. Therefore, the system will be developed as mobile first. (unless specified otherwise)

1. Constraints

No constraint identifies at this point.

1. Maintenance procedure

Let this document be V1.1 of a requirement specification document. Any changes hereafter must be logged into the change log found at the bottom of this document. The change logged shall reflect maintenance or changes to original specification that have already been implemented. Any technical or non-technical maintenance performed will be logged in by the change proposer or system design authority.

1. Future work and proofing

No future work identified at this point.

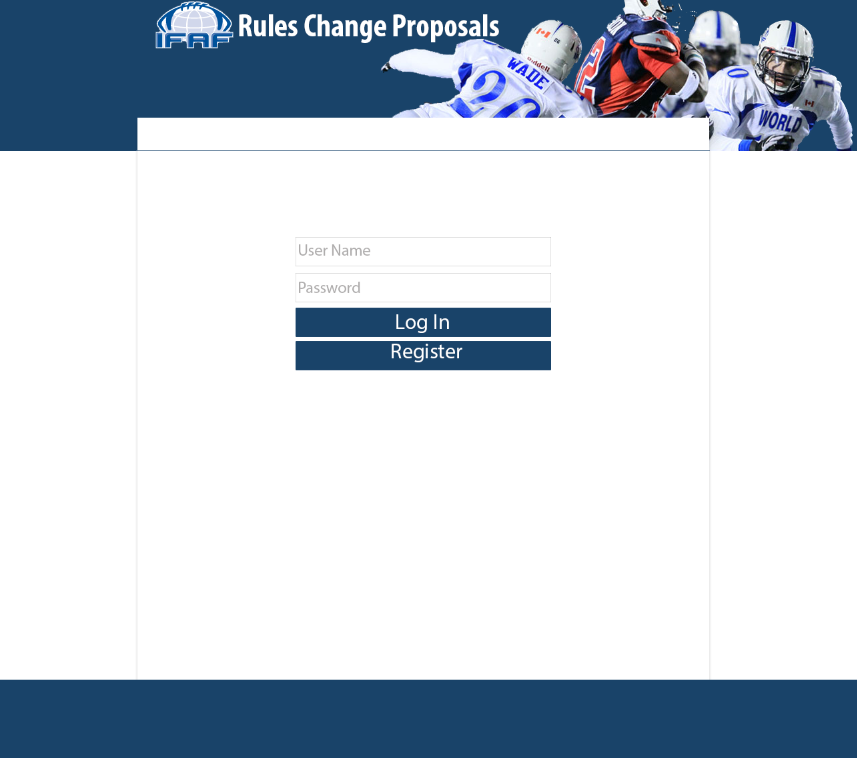
1. Change log

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change date** | **Changer** | **Changes** |
| **V1.1** | 29/02/2018 | Aseged W. | Requirement specification created |
| **V1.2** | 05/02/2018 | Aseged W. | Changed admin buttons position from left to right side of the view. |
| **V1.2** | 15/03/2018 | Aseged w. | Design constraint identified with jsf dataTable. Enable tabular view instead |

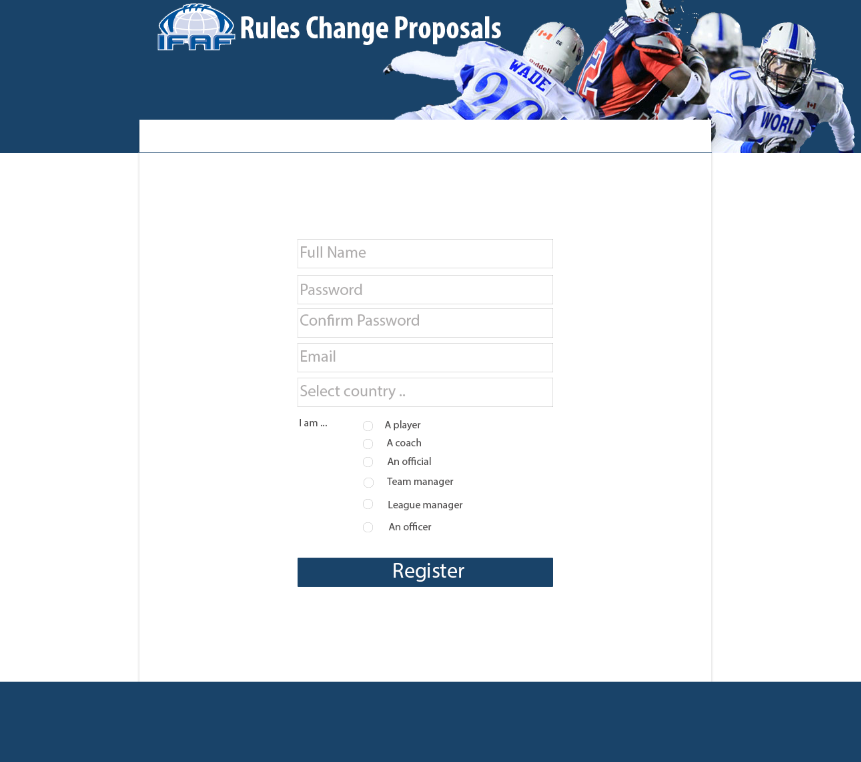
1. User Interface Design; First Iteration

As stated above, the application can be part of the existing IFAF website or a stand-alone application. Here, seen as the organization already have a website, and seen as no requirement have been specified regarding this, we are going to assume that the application will be part of the existing website.

The interface will be dynamic. This means we will use the same index page, however, we will hide or show additional functionalities depending on the user’s role. Once the user clicks on the link leading to the application, they will see either the login or registration page as follows;

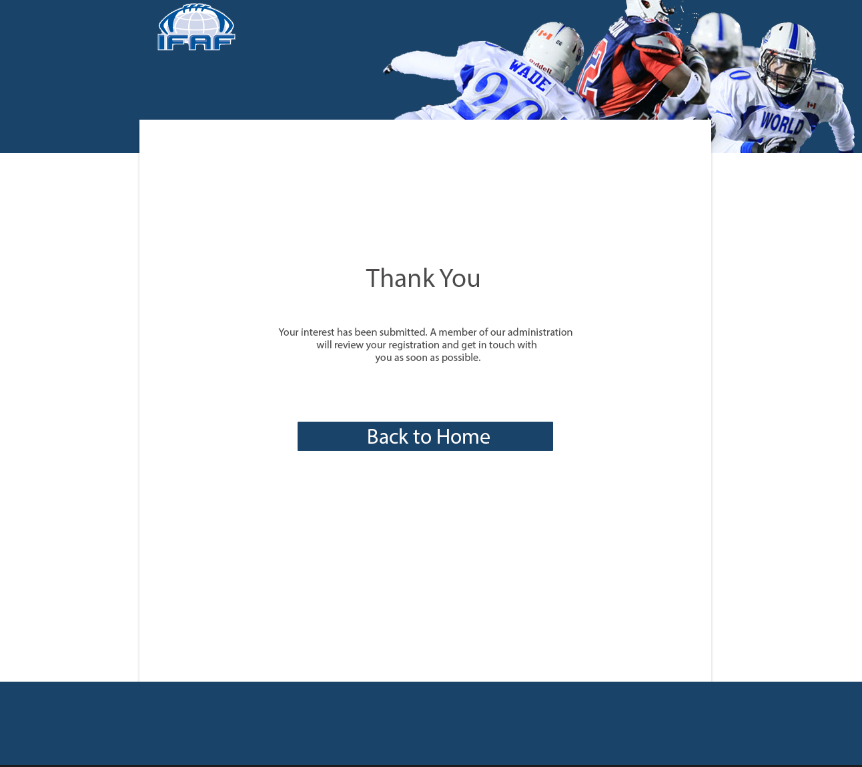


*Fig 1, Login page*



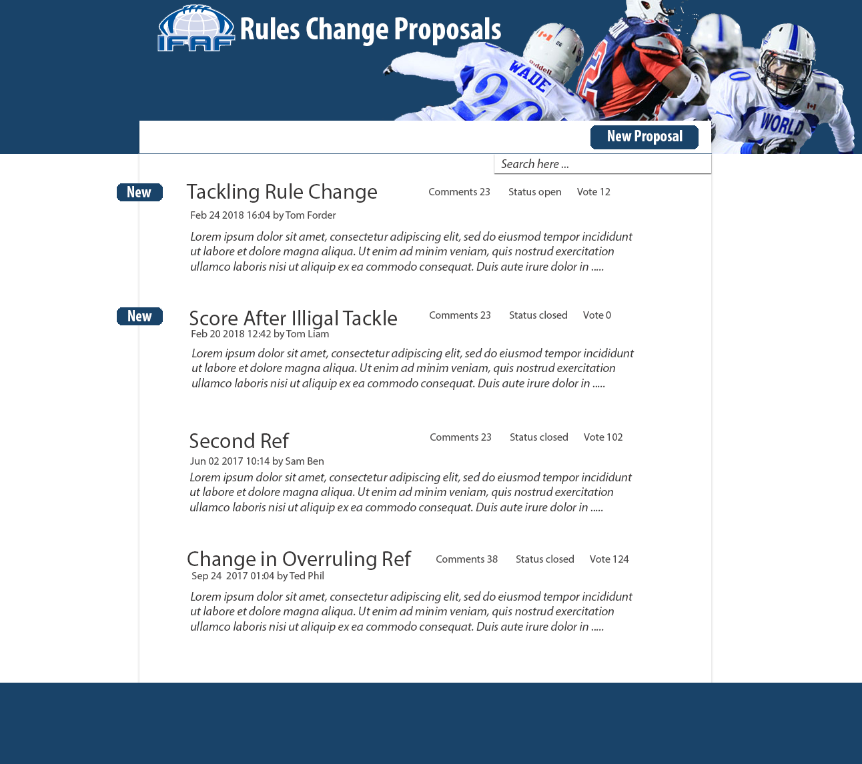
*Fig 2, Registration page*

Once users are registered, it will need to be approved by admin, therefore they will be redirected to a ‘Thanks You’ page where they will see instruction on what to do next.



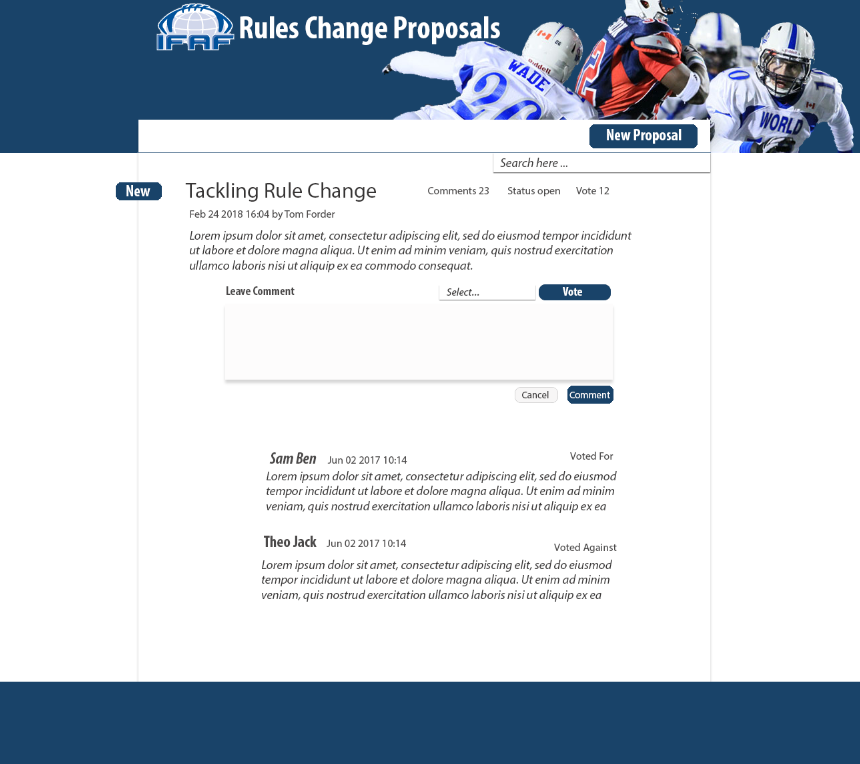
*Fig 3, Thank you page for completed registration*

Users who chose to log in will be redirected to the index page of the application, which will look like the following;



*Fig 4, Index page (Home or Landing page)*

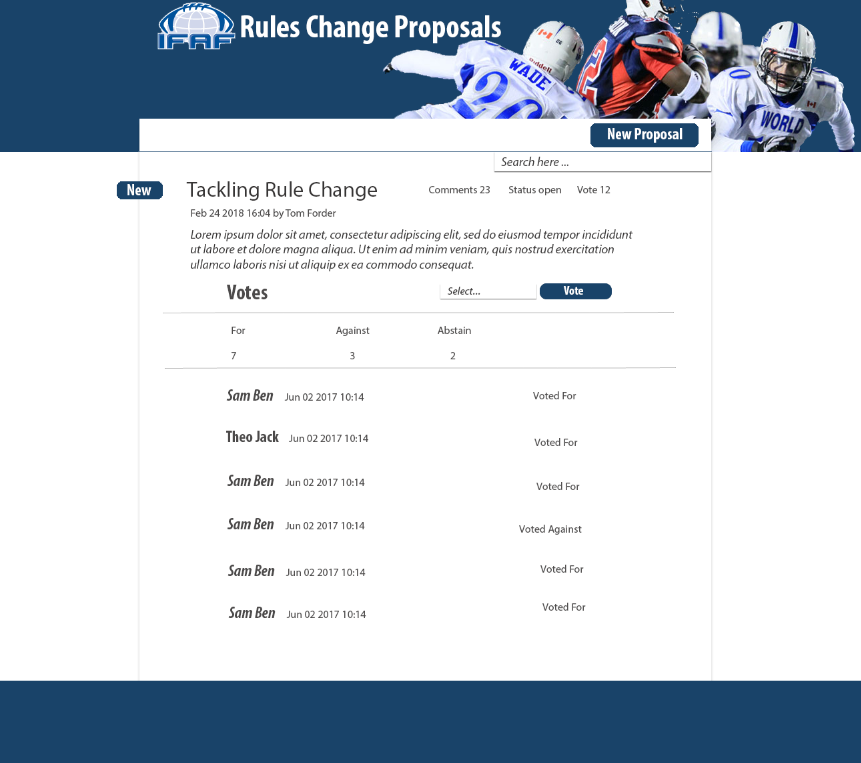
As shown above, the index (or home page) of the application has a social site style to it. It shows a list of all change proposals with time stamp, status and number of votes since voting is commenced. Users can click on the proposal itself to view details such as the full description of the proposal, full list of comments, full list of voters and their vote, as well as any variation of functionalities depending on the user role. (see 18.1) The detailed view of change proposals will look like the following;



*Fig 5, Proposal detail page*

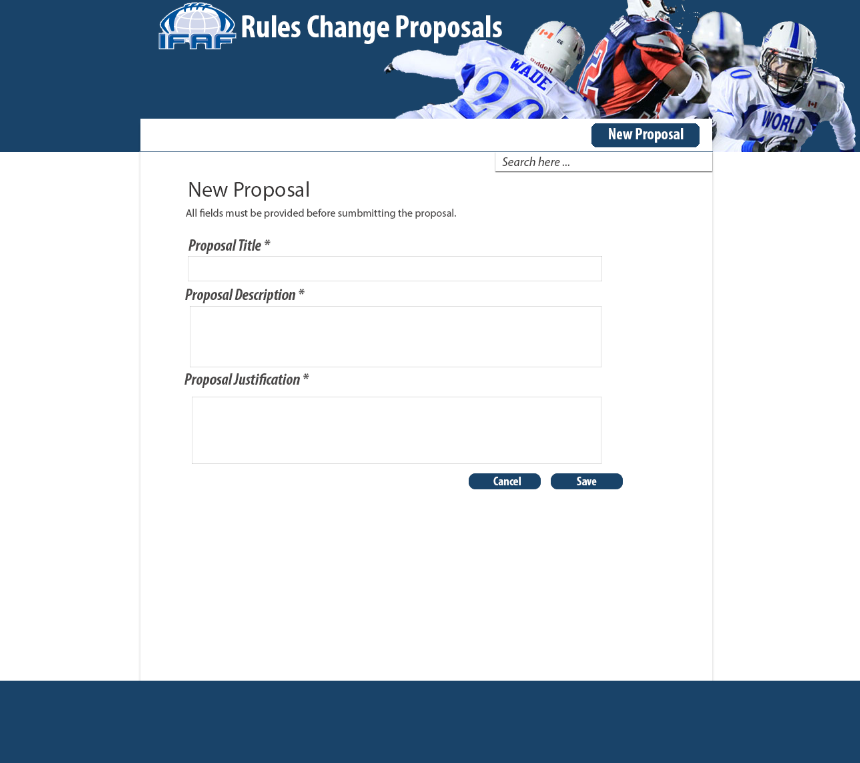
The functionality to comment or to vote will only be available to those with the role. Admin will see the same view but with extra functionalities to delete or hide comments or proposal.

From this page, admin and users who have access to the voting page will be able to click on the voting link to see a full list, which will look like the following;



*Fig 6, Proposal votes detail page*

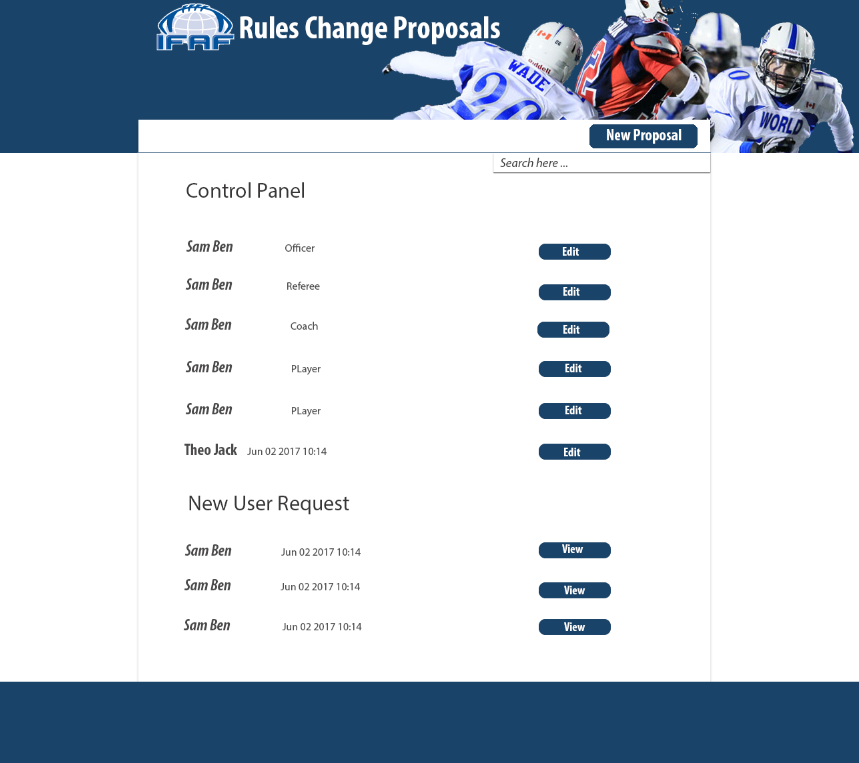
The index page will contain the ‘New Change Proposal’ button that will enable users to navigate to the new proposal submission page. The page will look like as follows;



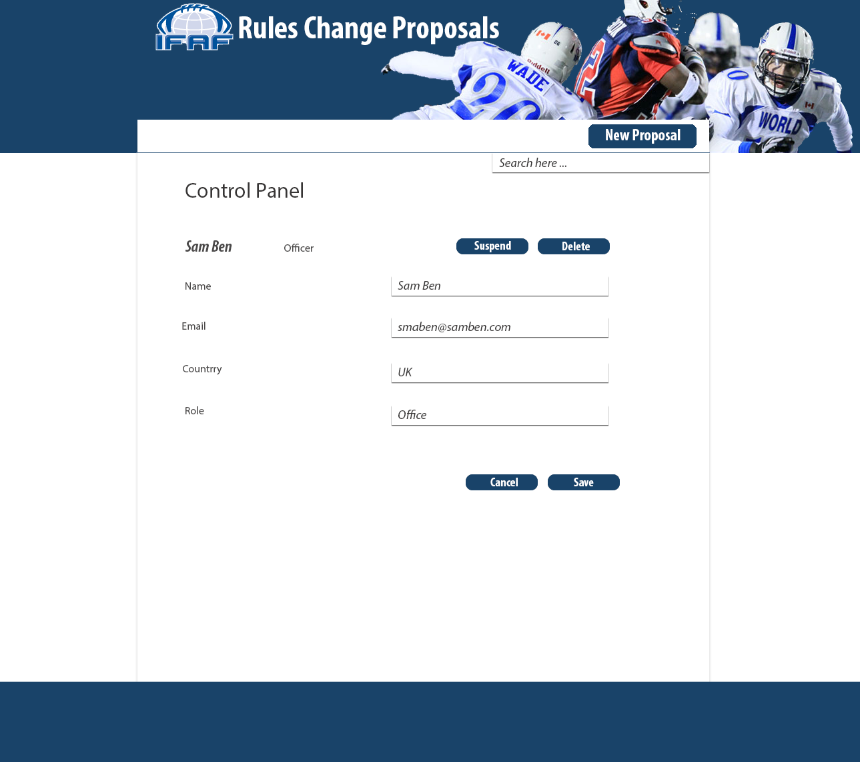
*Fig 7, New Proposal page*

18.1. Dynamic Functionalities

Some functionalities will simply be deactivated or hidden from the user. For instance, the Vote button and dropdown menu on the index page will only show if the user role includes voting. Another example is that if the proposal being viewed belongs to the user, then option for editing or withdrawing the proposal will be available. In the case of administrators, these functionalities will be available on all proposals and comments.



*Fig 8, Control panel page for user management*



*Fig 9, use editing page for administration*

1. Organizational Structure Diagram; First Iteration

The following organisational chart shows how the user will navigate through the views shown above in the user interaction design. Most of the view can be navigated to through the index page. There are some dedicated buttons that will be used for navigating to a specific control panel views. Otherwise, we will be using links on each view to navigate to corresponding views.

