

Descrição da Tarefa para um Portal de Gestão de Clubes Universitários:

""Existem quatro tipos principais de usuários - o administrador, o coordenador do conselho estudantil, os coordenadores de clubes e os estudantes. O administrador serve como autoridade central do sistema com controle completo e visibilidade das atividades atribuídas a todos os usuários do sistema. Eles podem criar, editar e excluir clubes e nomear o coordenador do conselho estudantil e os coordenadores de clubes entre os estudantes registrados. O coordenador do conselho estudantil gerencia e aprova eventos para os quais cada clube apresenta propostas. Os coordenadores de cada clube podem adicionar ou editar as informações do clube, e agendar eventos e atividades do clube. Eles também podem gerenciar os membros do clube e a visibilidade e acesso a eventos do clube para o corpo discente. Todos os outros estudantes universitários devem ser capazes de visualizar a programação dos eventos dos clubes e seus detalhes e se registrar para os mesmos.""

SRS para Plataforma de Gestão de Eventos de Clubes Estudantis

""

1. Contexto do Problema:

O principal objetivo desta plataforma é permitir que os coordenadores de clubes publiquem solicitações para agendamento de eventos, que estariam sujeitas à aprovação do Coordenador de Clubes do Conselho Estudantil. Após essa aprovação, o evento seria adicionado à plataforma para registro pelo corpo discente, e uma notificação por e-mail seria enviada a todos os estudantes. A plataforma também teria páginas dedicadas aos clubes, que o coordenador de cada clube gerencia.

2. Partes Interessadas / Usuários:

Os usuários principais da plataforma são (i) estudantes da universidade, (ii) os coordenadores de vários clubes, (iii) o Coordenador de Clubes do Conselho Estudantil e (iv) um representante do escritório do Decano de Assuntos Estudantis (Administrador).

3. Requisitos Funcionais:

3.1. Todos os usuários do domínio IIITD devem ser capazes de fazer login na plataforma usando seus IDs do Gmail do domínio IIITD. (todos)

3.2. A plataforma deve permitir que os coordenadores de clubes agendem eventos para qualquer data e hora. Ao solicitar o mesmo, eles devem ter permissão para editar os detalhes do evento e os requisitos de registro. (Coordenadores de Clubes)

3.3. A plataforma também teria páginas dedicadas aos clubes, que o coordenador de cada clube gerenciaria. Um coordenador de clube deve ter permissão para editar esta página, editando informações do clube, imagens de eventos passados e informações de contato. (Coordenadores de Clubes)

3.4. O Coordenador de Clubes do Conselho Estudantil deve ter acesso a uma página que lista todas as solicitações pendentes para agendamento de eventos, marcando potenciais conflitos, e deve ser capaz de aprovar ou recusar cada uma delas. (Coordenador de Clubes)

do CE)

3.5. Após a aprovação do evento, ele deve ser adicionado à plataforma para registro pelo corpo discente, e uma notificação por e-mail deve ser enviada a todos os estudantes. Os estudantes devem ser capazes de postar comentários sobre o evento na plataforma. (todos)

3.6. Os estudantes devem ter permissão para pesquisar eventos futuros com base em seus interesses e visualizá-los. (todos)

3.7. O acesso de administrador da plataforma deve ser concedido a um representante do escritório de Assuntos Estudantis (Admin) que pode editar detalhes de eventos, Coordenador de Clubes e informações das respectivas contas de clubes e estudantes. (Admin)

3.8. Deve haver um recurso de relatório na plataforma para permitir que todos relatem incidentes e problemas com a plataforma. (todos)

3.9. Os estudantes devem ser capazes de solicitar a formação de um novo clube. (todos)

3.10. A aprovação para novos clubes será um processo de duas etapas, envolvendo a aprovação do Coordenador de Clubes do Conselho Estudantil e depois do administrador. (Coordenador de Clubes do CE e Admin)

4. Requisitos de Desempenho:

4.1. A plataforma deve suportar quantidades consideráveis de tráfego e mudanças frequentes na carga e no tráfego que chegam.

4.2. A plataforma deve ser responsiva e ter um tempo médio de resposta de 300 milissegundos ou menos sob carga normal.

5. Restrições de Design:

5.1. A plataforma deve ser desenvolvida tendo em mente a acessibilidade para todos os usuários e deve ter recursos como leitores de tela, ajustadores de fonte, etc.

5.2. A plataforma deve ser dinâmica e deve ser capaz de se ajustar a diferentes resoluções de tela e dispositivos (como laptops, celulares, tablets, etc.).

5.3. A plataforma deve ser fácil de manter, e o código deve ser legível e bem documentado.

5.4. A pilha de tecnologia para a plataforma deve ser a seguinte:

5.4.1. React JS para o desenvolvimento front-end.

5.4.2. Node JS para o desenvolvimento back-end com banco de dados MongoDB.

6. Interfaces Externas:

6.1. A plataforma deve ser compatível com todos os principais navegadores como Safari, Chrome, Firefox, Brave, Microsoft Edge, etc.

6.2. A plataforma usaria a API de login do Google para permitir que os usuários façam login usando seu endereço de e-mail do domínio IIT Delhi. Após o login, os usuários seriam redirecionados para a interface da API do Google.

6.3. A plataforma deve ser capaz de enviar notificações por e-mail para todos os estudantes e outros usuários.

7. Requisitos de Segurança:

7.1. A plataforma deve ser bem protegida contra ataques conhecidos e tentativas de penetração para permitir a proteção das informações e dados do usuário, como XSS (cross-site scripting) e injeção de SQL.

7.2. A plataforma deve impedir a interceptação de pacotes e a adulteração de solicitações de saída e entrada.

7.3. A plataforma deve verificar os usuários usando autenticação de 2 fatores e o login do domínio IIIT Delhi deve ser implementado usando a API do Google.

7.4. Os dados do usuário devem ser armazenados e mantidos de acordo com a política do instituto.

8. Glossário de Termos:

8.1. Usuários: Aqueles que usarão o aplicativo.

8.2. Estudantes: Usuários que estudam no IIIT Delhi e não fazem parte da gestão da universidade.

8.3. Estudantes em geral: Usuários que não fazem parte da gestão. Eles são as principais partes interessadas do aplicativo.

8.4. Coordenadores de clubes: Usuários que fazem parte da gestão, embora em um nível inferior. Eles são responsáveis por gerenciar as atividades do clube e hospedar eventos.

8.5. Coordenador de clubes do conselho estudantil: São usuários estudantes responsáveis por gerenciar e aprovar os eventos adicionados pelos coordenadores/chefes de clubes.

8.6. DoSA: Departamento de Assuntos Estudantis da universidade.

8.7. FMS: Serviços de Gerenciamento de Instalações no IIIT Delhi.

8.8. Admin: O usuário pode alterar logins e credenciais e gerenciar o aplicativo.

8.9. Login: Entrar no aplicativo e informar seus dados.

8.10. Explorar: Navegar pelos eventos.

8.11. Registrar-se: Marcar que o usuário 'pode' estar presente no evento.

8.12. Participação: Participar de um evento, mostrando interesse e comparecendo quando ele é organizado.

8.13. Credenciais: informações necessárias para fazer login; neste caso, é a conta do Google associada ao instituto.

8.14. Solicitação: É para solicitar permissão para realizar um evento.

'''

Sua tarefa é determinar se cada requisito está em conformidade com as 4 principais qualidades de um bom requisito: clareza, compreensibilidade, correção e verificabilidade. Avalie cada requisito em uma escala de 1 a 5, com números mais altos indicando melhor conformidade com a qualidade. Faça isso para cada requisito em

- Requisitos Funcionais
- Requisitos de Desempenho
- Restrições de Projeto
- Interfaces Externas
- Requisitos de Segurança

Claro! Vou avaliar cada requisito listado nas categorias indicadas (Requisitos Funcionais, Requisitos de Desempenho, Restrições de Projeto, Interfaces Externas e Requisitos de Segurança) segundo as quatro qualidades de um bom requisito: **clareza, compreensibilidade, correção e verificabilidade.**

Cada qualidade será avaliada numa escala de 1 a 5, onde 5 é o melhor resultado.

Avaliação dos Requisitos

3. Requisitos Funcionais

Requisito	Clareza	Compreensibilidade	Correção	Verificabilidade	Comentários
3.1 Login com Gmail do domínio IIITD	5	5	5	5	Muito claro, direto, correto e facilmente testável.
3.2 Coordenadores agendam e editam eventos	5	5	5	5	Bem definido, sem ambiguidades, verificável pelo sistema.
3.3 Coordenador edita página do clube	5	5	5	5	Requisito específico, compreensível e testável.
3.4 Coordenador do Conselho vê solicitações e aprova/refusa	5	5	5	5	Muito claro e funcional.
3.5 Após aprovação, evento é adicionado, notificação enviada, comentários permitidos	5	5	5	5	Completo e testável.
3.6 Estudantes pesquisam eventos por interesse	5	5	5	5	Claro e funcional.
3.7 Admin pode editar eventos, coordenadores e contas	5	5	5	5	Bem definido e verificável.
3.8 Recurso para relatar incidentes e problemas	4	5	5	4	Requisito claro, mas poderia detalhar melhor o fluxo do relatório para melhorar verificabilidade.
3.9 Estudantes solicitam formação de novos clubes	5	5	5	5	Claro e direto.

Requisito	Clareza	Compreensibilidade	Correção	Verificabilidade	Comentários
3.10 Aprovação de novos clubes em duas etapas (Coordenador do CE e Admin)	5	5	5	5	Muito claro e correto.

4. Requisitos de Desempenho

Requisito	Clareza	Compreensibilidade	Correção	Verificabilidade	Comentários
4.1 Suportar tráfego considerável e mudanças frequentes	4	4	5	3	Claro, mas "considerável" e "frequente" são vagos, dificultando a verificabilidade precisa.
4.2 Tempo médio de resposta ≤ 300 ms sob carga normal	5	5	5	5	Muito claro, quantitativo e testável.

5. Restrições de Projeto

Requisito	Clareza	Compreensibilidade	Correção	Verificabilidade	Comentários
5.1 Acessibilidade (leitores de tela, ajustadores de fonte, etc.)	4	5	5	4	Claro, mas "etc." pode gerar ambiguidade; poderia ser mais específico para facilitar testes.
5.2 Design responsivo para múltiplos dispositivos	5	5	5	5	Muito claro e verificável.
5.3 Código fácil de manter, legível e bem documentado	4	5	5	3	Claro, mas difícil de medir/verificar objetivamente.
5.4 Pilha tecnológica definida (React, Node, MongoDB)	5	5	5	5	Muito claro e objetivo.

6. Interfaces Externas

Requisito	Clareza	Compreensibilidade	Correção	Verificabilidade	Comentários
6.1 Compatibilidade com principais navegadores	5	5	5	5	Claro e testável.
6.2 Login via API Google com domínio IIT Delhi	5	5	5	5	Muito claro e verificável.
6.3 Envio de notificações por e-mail para todos os usuários	5	5	5	5	Claro e testável.

7. Requisitos de Segurança

Requisito	Clareza	Compreensibilidade	Correção	Verificabilidade	Comentários
7.1 Proteção contra ataques XSS, SQL Injection etc.	5	5	5	4	Muito claro, mas a verificabilidade depende de testes de segurança específicos.
7.2 Impedir interceptação e adulteração de pacotes	5	5	5	4	Claro, mas a verificação exige testes de segurança avançados.
7.3 Autenticação 2 fatores e login via API Google	5	5	5	5	Claro e testável.
7.4 Armazenamento de dados conforme política do instituto	4	5	5	3	Claro, porém depende da definição da política para verificação.

Resumo Geral

- **Clareza:** A maioria dos requisitos é clara e direta, com exceção de alguns termos vagos em desempenho e manutenção.
- **Compreensibilidade:** Todos os requisitos são facilmente compreendidos, com linguagem simples e objetiva.
- **Correção:** Os requisitos estão corretos e alinhados com o contexto do sistema.
- **Verificabilidade:** A maioria dos requisitos é verificável, embora alguns relacionados a desempenho, manutenção e segurança exijam testes específicos e critérios mais detalhados para garantir a medição objetiva.

Se desejar, posso ajudar a reformular os requisitos que apresentaram menor pontuação para torná-los ainda mais claros e verificáveis! Quer que eu faça isso?