SKPL-xxxx

SOFTWARE REQUIREMENTS SPESIFICATION

VIE

for:

RPLGDC Laboratory

Prepared by:

1. Dzaki Mahadika Gunarto (SID 1301192286)
2. Dharaka Pranamya Mahadhika Damar (SID 1301194248)
3. Muhammad Ilham Mubarak (SID 1301194276)

Informatics Major – School of Computing

Telkom University

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

Daftar Perubahan

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX	-	A	B	C	D	E	F	G
TGL								
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Table of Contents

1. Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose of this document is described and a list of abbreviations and definitions is provided

1.1 Document Purpose

The purpose of this document is to explain in detail about an internal team finder for Telkom University's Informatics laboratory. It will explain the features, functional requirements, non-functional requirements, and the constraint for the system. This document is intended as documentation and references for future development.

1.2 Document Conventions

Bold text – Signifies important content or keywords.

1.3 Product Scope

Vie is a website that can find/create a team between Telkom University's Informatics laboratory. Its purpose is to help Telkom University Informatics student find their team to compete in a competition. The current service that provides similar functionality is Faceit. Faceit can create a team based on their accomplishment/tournament history but it is for a game purpose like CSGO, DOTA 2, etc. Vie proposes to create a simple website that can match people between different labs, help Telkom University Informatics students to get information about any competition, and also compete in some competition.

1.4 References

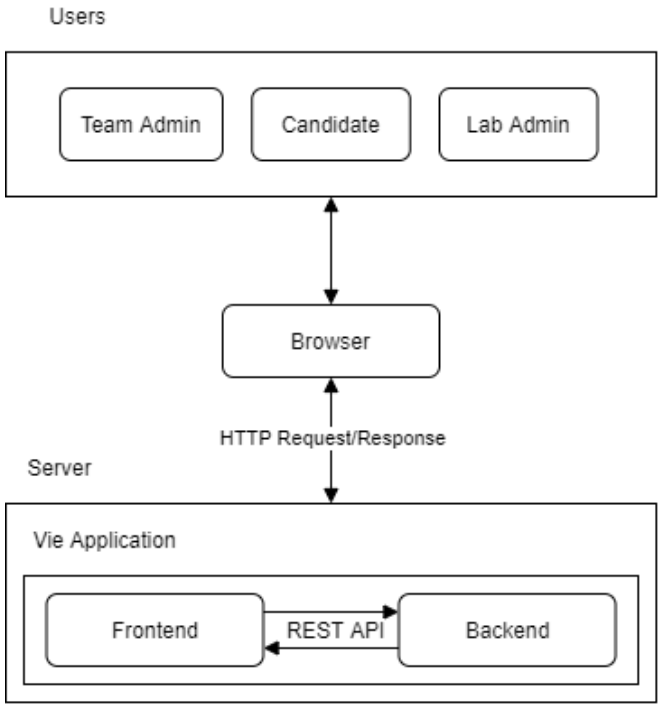
- Material Design (https://material.io/design)
- Tailwindcss Breakpoints (https://tailwindcss.com/docs/breakpoints)
- Web Vitals (https://web.dev/vitals)
- ISO/IEC 25010:2011 (https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en)

2. Overall Description

2.1 Product Perspective

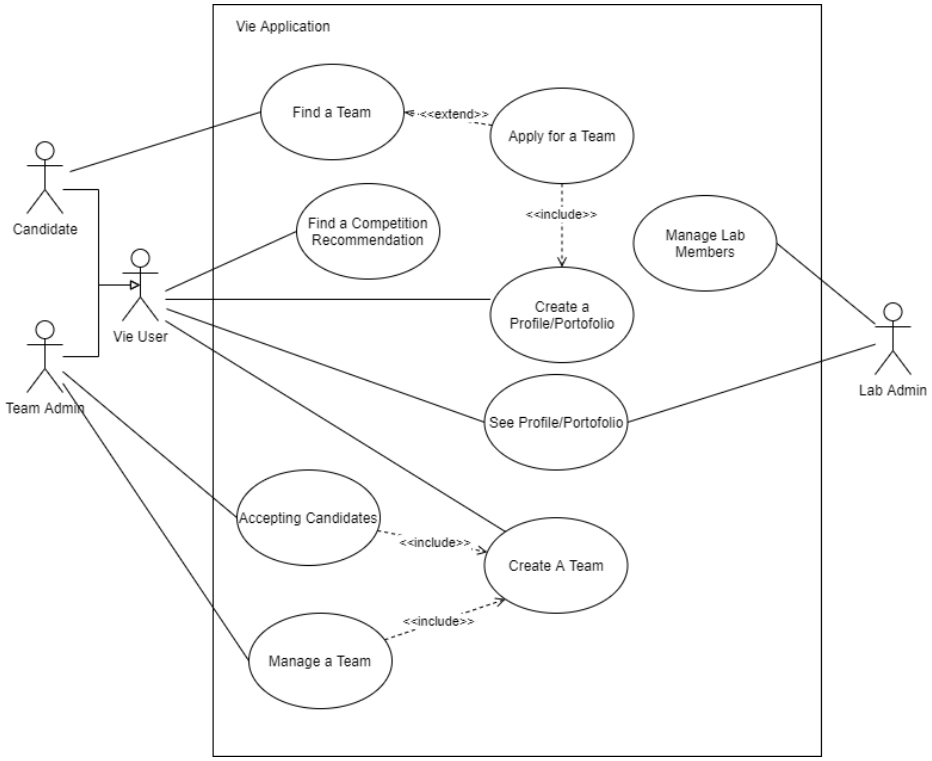
There is currently no student-made website to find or create a team for Informatics Faculty students at Telkom University. Existing solutions to find or create a team is a direct invitation by face-to-face meeting or using personal chat to the other student. This means they are required to have more effort in a face-to-face meeting or finding the other student contact to do a personal chat. The popular team finding website is FACEIT, however, FACEIT is a team finding website focusing on E-Sport tournament, while ours is a general competition.

The uses of team finding are limited. However, there are several interesting uses of team finding, for example, a user wants to find or create a team to compete in a competition, a user wants to create a project and need a team. Members of a team are limited depends on their own team preferences.



From the diagram above, our general idea of the application is a web-based application that can be opened in almost all web browsers. We have 3 types of user that will be explained in more detail in the next section. For the web application itself, Vie is using a frontend with a backend using REST API deployed in a server.

2.2 Product Functionality



The use case above shows a general overview of the product functionality. On a larger scale, the application has these 3 main functionalities.

- A user can go to the website and find or create a team based on their competencies or a competition.
- A user can go to the website and look for a competition recommendation.
- A user can go to the website and create their own profile.
- A user can go to the website and create a team based on competition.

A more in-depth explanation of the scenarios will be explained in chapter 4.

2.3 Classes and Characteristics Users

Name	Description
Team Admin	Team admin is a user who manages their own team. Team admin able to create a team and find members to fulfill their team, able to accept the candidate's applications, and able to change the maximum member of their team. Team admin able to create their profile or portfolio and see the candidate's profile and portfolio and also able to see the competition recommendation to compete.
Candidate	A candidate is a user who needs a team. A candidate is able to search and join a team to compete in a competition. A candidate is unable to accept other candidate's applications and unable to change the maximum member of their team. A candidate is able to create a team and become a team admin. A candidate is able to create their profile or

	portfolio and see another candidate's profiles and portfolios and also able to see the competition recommendation to compete.
Lab Admin	Lab Admin is a user who manages their own laboratory. Lab admin is able to create and edit their laboratory's profile. Lab Admin is able to create and change their laboratory's members' data. Lab Admin is able to see the candidate's profiles and portfolios.

2.4 Operating Environment

Our application is a web-based application therefore the operating environment for the user is as follows.

- Web Browser (Recommended latest Chrome).
- Windows, macOS, Linux, iOS, Android.

2.5 Planning and Implementation Constraints

2.5.1 Design

Our application follows the design guideline from **Material Design**. All of the layouts, typography, accessibility guideline, and more follows their principles. However, in terms of breakpoints, the application follows **Tailwindcss** breakpoints as they are based on the common device resolutions.

sm	640px
md	768px
lg	1024px
xl	1280px
2xl	1536px

The breakpoints are used as the main references when working on media queries.

2.5.2 Software

Library	**Version**
React	^16.0
NodeJS	^12.0
Express	^4.0
PostgreSQL	^13.1

2.6 User Documentation

Our website will be focusing on user-friendly experiences. The website will have effective navigation to help users while accessing our website. In addition, a live navigation tutorial on our website will be made to help users while accessing our website for the first time.

2.7 Assumptions and Dependecies

Assumptions	**Fact**
The user does not need to see their friends list	The user shall be able to access and see their friends list
The user does not need to receive a notification when there is a new competition recommendation	The system shall notify the user each time new competitions are available
The user does not need to communicate with their teammate using a voice channel	The system shall be able to provide a voice channel as a way to communicate with the team members

3. External Interface Requirements

3.1 User Interface

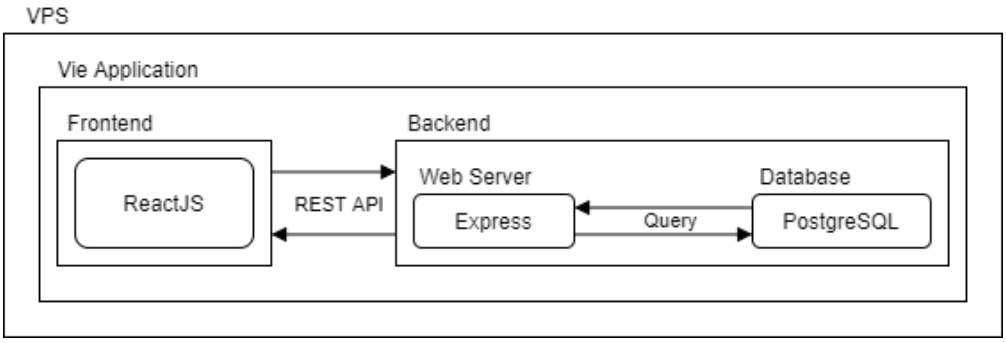
The UI Interface follows the design system from the previous chapter with some additional requirements below.

- The “primary” color used is #E71D36.
- The “primary-text” color used is #FFFFF.
- The font family used is “Source Sans Pro”.
- The application supports light and dark mode.

As for the UX, the requirements are defined below.

- The system shall be able to provide a voice channel as a way to communicate with the team members.
- The system shall notify the user each time new competitions are available.
- The system shall be able to filter teams by team name and/or by competition.
- The team admin shall be able to change the maximum number of a team.

3.2 Software Interfaces



Vie Application consists of **ReactJS** as the frontend UI library, NodeJS with **Express** framework as the backend to make the REST API, **PostgreSQL** as the DBMS. The application is deployed to a VPS.

3.3 Communication Interfaces

The main communication protocol that we used is HTTP/HTTPS.

4. System Features

This section illustrates how to set functional requirements for products based on system features, the main services provided by the product.

4.1 Finding a Team

4.1.1 Description:

This is the main feature of our application. The candidate can find a team and apply for the team if they are interested.

4.1.2 Trigger:

The candidate needs to go to the homepage and search the team on the search bar.

4.1.3 Input:

The input is the team name or competition name.

4.1.4 Output:

The output is a list of teams that is available based on the candidate’s search query.

4.1.5 Main Scenario:

Assume that the candidate is registered, they can access the search bar on the homepage and they can search for the team by name or by competition. They will be redirected to a page that shows a list of all available teams based on the search query. They can view team information. If they are interested to join, they can apply to the team with their portfolio and wait for approval from the team admin. After the approval, they became a member of that team.

4.1.5.1 Preconditions: The candidate is registered and at the homepage.

4.1.5.2 Post-conditions: The candidate see a list of teams and joined a team.

4.1.5.3 Step by Step:

- The candidate is at the homepage.
- The candidate fill in the search bar.
- The candidate see a list of teams that they have searched.
- The candidate opens a team.
- The candidate applies for the team.
- The candidate accepted to the team.

4.1.6 Exceptional Scenario 1:

Assuming that the candidate is registered, they can access the search bar on the homepage and they can search for the team by name or by competition. The query resulting in no teams. They will be redirected to a page that shows no teams available.

4.1.6.1 Preconditions: The candidate is registered and at the homepage.

4.1.6.2 Post-conditions: The candidate see a warning message saying there is no team available.

4.1.6.3 Step by Step:

- The candidate is at the homepage.
- The candidate fill in the search bar.
- The query resulting in no team.
- The candidate sees a warning message.

4.1.7 Exceptional Scenario 2:

Assume that the candidate is registered, they can access the search bar on the homepage and they can search for the team by name or by competition. They will be redirected to a page that shows a list of all available teams based on the search query. They can view team information. If they are interested to join, they can apply to the team with their portfolio and wait for approval from the team admin. The team admin rejects the candidate.

4.1.7.1 Preconditions: The candidate is registered and at the homepage.

4.1.7.2 Post-conditions: The candidate is rejected by the team admin.

4.1.7.3 Step by Step:

- The candidate is at the homepage.
- The candidate fill in the search bar.
- The candidate see a list of teams that they have searched.

- The candidate opens a team.
- The candidate applies for the team.
- The candidate rejected from the team.

4.2 Competition Recommendation

4.2.1 Description:

This feature can show the list of competition recommendations for every user to make it easier for the users to find information about any competitions that available.

4.2.2 Trigger:

The user needs to go to the homepage to show the list of competition recommendations.

4.2.3 Input:

There is no input in this feature because the competition recommendations are auto showed to every user on the home page.

4.2.4 Output:

The output of this feature is the list of competition recommendations.

4.2.5 Main Scenario:

Assume that the user is registered, the user will open our website (after they logged in), and then the user can immediately see the list of competition recommendations on the home page.

4.2.5.1 Preconditions: The user has not logged in to our website.

4.2.5.2 Post-Conditions: The user can immediately see the list of competition recommendations on the home page.

4.2.5.3 Step by Step:

- The user will go to the home page after they logged in.
- The user can immediately see the list of competition recommendations on the home page.

4.2.6 Exceptional Scenario 1:

Our exceptional scenario is when there is no competition is coming anytime soon.

4.2.6.1 Preconditions: The user has not logged on to our website.

4.2.6.2 Post-Conditions: The user can immediately see the text “There is no competition coming anytime soon.” on the home page.

4.2.6.3 Step by Step:

- The user will go to the home page after they logged in.

- The user can immediately see the text “There is no competition coming anytime soon.” on the home page.

4.3 Profile to Show Portfolio

4.3.1 Description:

This feature can show the individual portfolio of a person such as their achievements, their recent competition, and their recent teams. This feature is available for every user.

4.3.2 Trigger:

To be executed, the user needs to go to the profile of a person, in the profile page, the user should click the “Show Portfolio” button to show the portfolio of that person.

4.3.3 Input:

The input of this feature is a request from the user to open the “Show Portfolio” page in someone’s profile page.

4.3.4 Output:

The output of this feature is someone’s portfolio.

4.3.5 Main Scenario:

Assume that the user is registered, the user will go to the profile of a person, then the user will see the profile page, after that the user should click the “Show Portfolio” button in the profile tab to show the portfolio of that person, and then the someone’s portfolio will appear in the user screen.

4.3.5.1 Preconditions: Team admin is on the profile page.

4.3.5.2 Post-Conditions: Someone’s portfolio will appear in the user screen.

4.3.5.3 Step by Step:

- The user will go to the profile of a person.
- The user will see the profile page.
- The user should click the “Show Portfolio” button in the profile tab.
- Someone’s portfolio will appear on the user screen.

4.3.6 Exceptional Scenario 1:

Our exceptional scenario is when the candidates do not have a portfolio, so their portfolio page is blank.

4.3.6.1 Preconditions: The user is on the profile page.

4.3.6.2 Post-Conditions: The user will see the “This candidate does not have a portfolio” text.

4.3.6.3 Step by Step:

- The user will go to the profile of a person.
- The user will see the profile page.
- The user should click the “Show Portfolio” button in the profile tab.
- The user will see the “This candidate does not have a portfolio” text.

4.4 Creating a Team

4.4.1 Description:

The team admin can create a team and can accept/reject candidates that apply for the team.

4.4.2 Trigger:

The candidate needs to go to the team page and create a team.

4.4.3 Input:

The input is the team name, the competition, number of maximum members, and the needed member’s roles.

4.4.4 Output:

The output is a new team created that can be managed by the team admin.

4.4.5 Main Scenario:

Assume that the team admin is registered, they can access the team page and click the “Create team” action. They will fill in the required form consisting of a team name, the competition, number of maximum numbers, and the needed member’s roles.

4.4.5.1 Preconditions: The team admin is registered and at the team page.

4.4.5.2 Post-conditions: The team admin successfully created a new team.

4.4.5.3 Step by Step:

- The team admin is on the team page.
- The team admin click the “Create Team” action.
- The team admin fill in the required forms.
- The team admin submits the forms.
- The team admin has successfully created a new team.

4.4.6 Exceptional Scenario 1:

Assume that the team admin is registered, they can access the team page and click the “Create team” action. They will fill in the required form consisting of a team name, the competition, number of

maximum numbers, and the needed member's roles, but they did not fill the team name so the error message appears on their screen.

4.4.6.1 Preconditions: The team admin is registered and at the team page.

4.4.6.2 Post-conditions: The team admin will see the "You must fill the team name" text.

4.4.6.3 Step by Step:

- The team admin is on the team page.
- The team admin click the "Create Team" action.
- The team admin fill in the required forms but they did not fill the team name.
- The team admin submits the forms.
- The team admin will see the "You must fill the team name" text.

5. Non-functional Requirements

5.1 Quality Attribute

For the whole application, we are following the quality attribute defined in ISO/IEC 25010:2011 (https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en) for the general application. As for the website itself, the website follows the web vitals metrics as defined in (https://web.dev/vitals/). These metrics can be used as a standard to measure the website user experience. Our non-functional requirements are as follows.

5.1.1 Availability

- Vie should be up and available 24hours.

5.1.2 Compatibility

- Vie should be able to run on any browser.

5.1.3 Performance Efficiency

- Any user interaction with Vie should not exceed 4 seconds.
- Vie should have the Largest Contentful Paint under 2.5 seconds.
- Vie should have the First Input Delay under 200 milliseconds.

Appendix A: List of Difficult Words

Words	Definition
Frontend	Frontend refers to the user interface that is implemented. The user interacts and sees the frontend using the web browser.
Backend	Backend refers to the webserver where the business logic, models, and database is implemented.
VPS	Stands for Virtual Private Server
API	Stands for Application Programming Interfaces
Express	Express is a minimal and flexible Node.js web application framework.
NodeJS	NodeJS is a JavaScript runtime built on Chrome's V8 JavaScript engine.