Introduction to Software Engineering

Requirements Analysis

The student team is required to complete the Software Requirements Specification (SRS) document for the assigned course project, following the attached template.

Software Engineering Department
Faculty of Information and Technology
University of Science

Table of Contents

Objectives 1

1 Member Contribution Assessment 2

2 Problem Statement 3

3 Requirements Overview 4

4 Requirements Analysis 5

5 Prototype/Mockup 6

Software Requirements Specification

Objectives

This document focus on the following topics:

- ✔ Complete the Software Requirements Specification (SRS) document with the following contents:
  - ▪ Elaborate on the Problem Statement
  - ▪ Overview of Requirements (Functional and Non-Functional), Stakeholders
  - ▪ Use Case Model
  - ▪ Use Case Specifications
  - ▪ Create Prototype and Mockup Diagrams of the System Interface
- ✔ Đọc hiểu tài liệu phân tích yêu cầu.

1 Member Contribution Assessment

ID	Name	Contribution (%)	Signature
23127006	Trần Nguyễn Khải Luân	20%	
23127113	Nguyễn Trần Phú Quý	20%	
23127144	Đinh Đại Vũ	20%	
23127179	Nguyễn Bảo Duy	20%	
23127189	Trần Trọng Hiếu	20%	

2 Problem Statement

Problem Statement

1. Introduction

In the rapidly expanding world of online gaming and esports, players rely increasingly on data analytics to enhance their gameplay, understand strategies, and stay competitive. One of the most popular strategy-based games today, *Teamfight Tactics (TFT)*, continuously evolves with new patches, units, and compositions. To remain competitive, players need accurate and up-to-date information about the current "meta," unit performance, and optimal team compositions.

However, this information is often fragmented across various online sources, requiring players to manually gather and analyze data. Many existing platforms present static or outdated data, lack intuitive interfaces, or fail to visualize trends in a user-friendly way.

This project aims to address these issues by developing a **web-based analytical platform**, similar in concept to *MetaTFT*, which provides real-time statistical insights, composition recommendations, and visualization of gameplay trends. The system will combine strong data analytics with an accessible user interface to help players make informed strategic decisions.

4. Scope and Objectives

The main goal of the project is to design and implement a data-driven web platform that helps users analyze and visualize the Teamfight Tactics meta environment effectively. The objectives include:

- Developing a responsive, intuitive front-end interface for data exploration and visualization.
- Implementing a scalable back-end API capable of handling real-time requests efficiently.
- Designing a normalized database schema to store and query statistical data.
- Ensuring secure and efficient communication between client, server, and database.

5. Expected Outcome

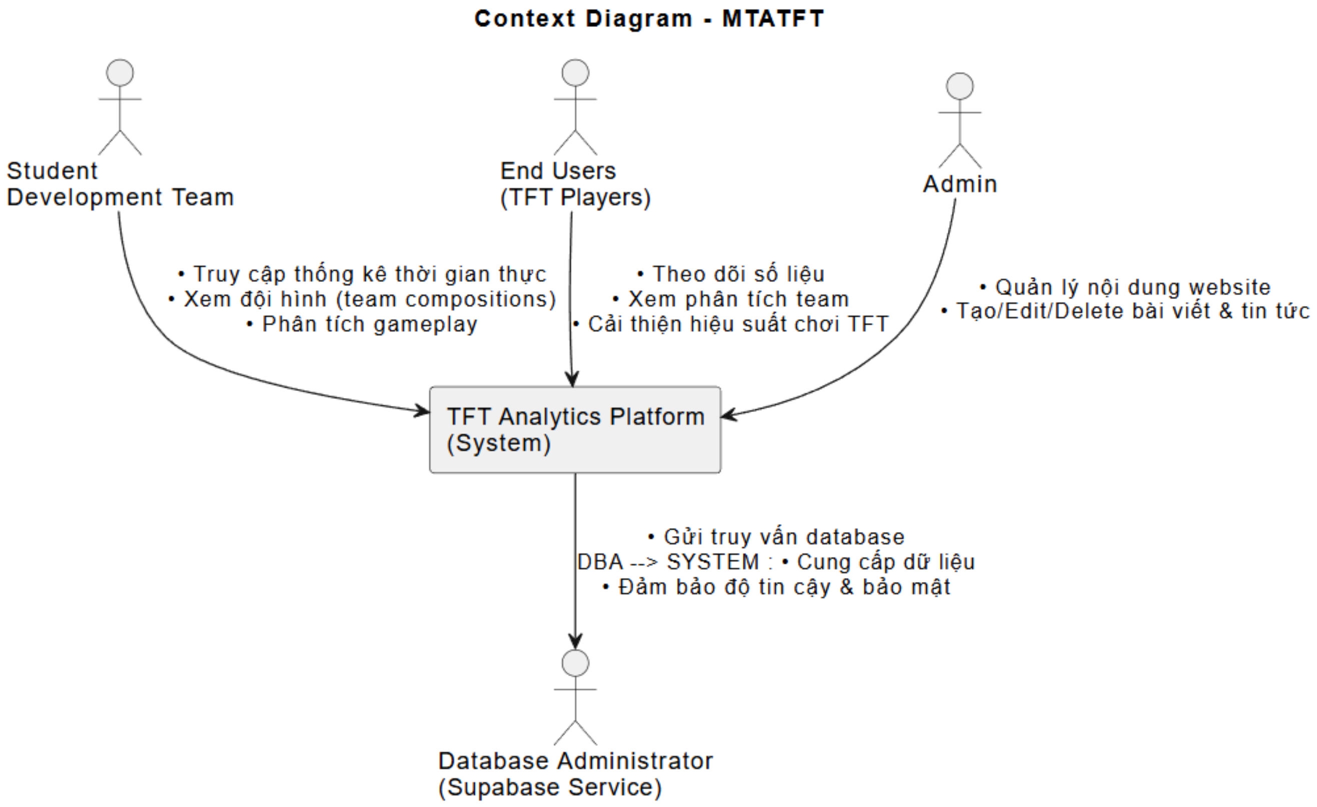
Upon completion, the project will deliver a fully functional **web-based analytics platform** for *Teamfight Tactics*. Users will be able to view game statistics, analyze top-performing compositions, and gain insight into trends across different patches and ranks.

The final product will demonstrate the application of full-stack development principles, cloud-based database management, and modern deployment techniques. It will also serve as a valuable example of combining data analytics and interactive web design to deliver real-time insights in a competitive gaming context.

3 Requirements Overview

3.1 Stakeholders

Image Link


Context Diagram - MTATFT
Student
Development Team
End Users
(TFT Players)
Admin
• Truy cập thống kê thời gian thực
• Xem đội hình (team compositions)
• Phân tích gameplay
• Theo dõi số liệu
• Xem phân tích team
• Cải thiện hiệu suất chơi TFT
• Quản lý nội dung website
• Tạo/Edit/Delete bài viết & tin tức
TFT Analytics Platform
(System)
• Gửi truy vấn database
DBA --> SYSTEM : • Cung cấp dữ liệu
• Đảm bảo độ tin cậy & bảo mật
Database Administrator
(Supabase Service)

Stakeholder	Description
Student Development Team	Primary users of the platform who access real-time statistics, team compositions, and performance analytics to improve their gameplay.

End Users (TFT Players)	Primary users of the platform who access real-time statistics, team compositions, and performance analytics to improve their gameplay.
Database Administrator (Supabase Service)	Manages data storage, real-time updates, and ensures the reliability and security of the PostgreSQL cloud database.
Admin	Manages website content such as posts, updates, and news. Has privileges to create, edit, and delete content to keep information accurate and up to date.

3.2 Requirements

3.2.1. Functional Requirements Specification

Id	***Group***	***Stakeholders' Needs***	***User Requirements***
R1	End Users	I want to know which team compositions are currently the strongest (meta) for climbing ranks.	Check the **Top Comps** table – it shows the strongest team compositions based on the latest data.
R2	End Users	I want to know detailed information about Champions, Items, and Upgrades.	Check **Game Stats** – provides detailed information about champions, items, and upgrades.
R3	End Users	I want to see my match history and ranking (and other players')	**Player Information Lookup** – displays information, match history, and

			player rankings.
R4	End Users	I want a Team Builder so I can try building a team.	**Team Builder** – allows users to create and test their own team compositions
R5	End Users	I want to save favorite teams to review later	**Capture team -** allows users to store their team by using pictures
R6	End Users	I want to post to discuss issues in the game	**Discussion posting** – users can upload images and write captions
R7	End Users	I want to be able to interact and comment on posts.	**Reactions and Comments** feature
R8	End Users	I want an account to post or save teams	**Login/Sign-up**, password encryption, and personal profile management
R9	End Users	I want to follow the post history of my favorite users.	**View Profile** of the account
R10	End Users	I want to customize my personal information	**Customize profile** – edit name, password, bio
R11	Admin	I want to moderate user-submitted content.	Admin can **review, edit, or delete** violating posts

R12	Admin	I want to manage user accounts.	Admin **manages users** (lock/unlock accounts)

3.2.2. Non-Functional Requirements Specification

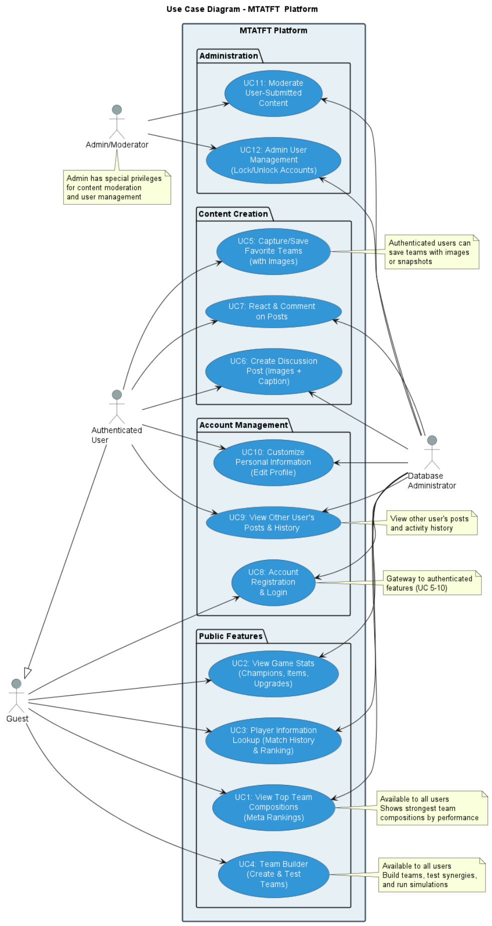
Id	***Group***	***Stakeholders' Needs***	***User Requirements***
NR1	End Users	I want the system to handle responses quickly.	Data response **< 1 minute**, optimized queries
NR2	End Users	I want the system to operate stably with few errors	Ensure clear testing and DevOps processes before release
NR3	Riot Games	I want Riot API data to be used correctly, ensuring fairness for players	Comply with Riot's API policies
NR4	End Users	I want to ensure that account information is managed securely and not exposed	Database login credentials, API keys, and tokens must be stored securely using environment variables, and must not be written directly in source code or version control systems.

NR5	Supabase	We want data to be used properly and transparently.	Ensure that information stored and used in the system is transparent.
NR6	End Users	I do not want my data or account to be leaked during usage.	The system must secure the connection between the Backend and Supabase using SSL/TLS encryption**.**

4 Requirements Analysis

4.1 Use Case model

Image Link


Use Case Diagram - MTATFT Platform
MTATFT Platform
Administration
UC11: Moderate User-Submitted Content
UC12: Admin User Management (Lock/Unlock Accounts)
Admin/Moderator
Admin has special privileges for content moderation and user management
Content Creation
UC5: Capture/Save Favorite Teams (with Images)
Authenticated users can save teams with images or snapshots
UC7: React & Comment on Posts
UC6: Create Discussion Post (Images + Caption)
Authenticated User
Account Management
UC10: Customize Personal Information (Edit Profile)
Database Administrator
UC9: View Other User's Posts & History
View other user's posts and activity history
UC8: Account Registration & Login
Gateway to authenticated features (UC 5-10)
Public Features
UC2: View Game Stats (Champions, Items, Upgrades)
Guest
UC3: Player Information Lookup (Match History & Ranking)
UC1: View Top Team Compositions (Meta Rankings)
Available to all users
Shows strongest team compositions by performance
UC4: Team Builder (Create & Test Teams)
Available to all users
Build teams, test synergies, and run simulations

4.2 Use Case Specification

4.2.1. Use Case 1

Use Case ID	U001
Use Case	View Top Team Compositions (Top Comps)
Brief Description	End user views the current strongest team compositions (meta) ranked by performance using the latest data.

Actor	End User (visitor or authenticated user), Database Administrator
Pre-Condition	The system has up-to-date composition statistics (aggregated from match data). The user has network access.
Result	Users see a ranked Top Comps table with composition name, win-rate, pick-rate, trend indicator, sample champions/items, patch and rank filtering.
Main Scenario	1. The user opens the Top Comps page. 2. The system fetches data. 3. System displays Top Comps. 4. User views or filters compositions.
Alternative Scenarios	A1: No recent data → show message. A2: Backend failure → retry option.
Non-Functional Constraints	Response ≤ 1m. Data freshness shown. Scalable and mobile responsive.

4.2.2. Use Case 2

Use Case ID	U002
Use Case	View Game Stats (Champions, Items, Tools)

Brief Description	User views detailed metadata and aggregated usage statistics for champions, items, and upgrades.
Actor	End User, Database Administrator
Pre-Condition	The system stores canonical metadata and aggregated metrics.
Result	The user receives detailed pages with description, metrics, builds, and synergies.
Main Scenario	1. The user selects a champion/item. 2. The system retrieves and displays detailed data. 3. User views builds and trends.
Alternative Scenarios	A1: Entity not found → show suggestions. A2: Data unavailable → display metadata only.
Non-Functional Constraints	Latency ≤ 300ms for metadata. Cached responses. Responsive layout.

4.2.3. Use Case 3

Use Case ID	U003
Use Case	Player Information Lookup (Match History & Ranking)

Brief Description	The user looks up a player to view match history, ranking and stats.
Actor	End User, Database Administrator
Pre-Condition	Player data available; system connected to match database.
Result	Player profile with rank, matches, and performance trend displayed.
Main Scenario	1. The user searches for the player name. 2. The system retrieves data. 3. User views match.
Alternative Scenarios	A1: Player not found → show suggestions. A2: Profile private → show limited info.
Non-Functional Constraints	Query latency ≤ 2s. Privacy compliance enforced.

4.2.4. Use Case 4

Use Case ID	U004
Use Case	Team Builder — Create & Test Teams
Brief Description	User builds and evaluates team compositions, tests synergies, and runs simulations.
Actor	End User

Pre-Condition	Champion/item metadata available; user has access to Team Builder UI.
Result	Users receive team evaluation metrics and can save or share the team.
Main Scenario	1. User opens Team Builder. 2. The system updates synergy scores in real-time. 3. User saves or tests team.
Alternative Scenarios	A1: Invalid team → show validation. A2: Simulation queue busy → retry option.
Non-Functional Constraints	Real-time response ≤ 1s. Computation limited for scalability.

4.2.5. Use Case 5

Use Case ID	U005
Use Case	Capture / Save Favorite Teams
Brief Description	An authenticated user saves favorite team compositions with images or snapshots.
Actor	End User (authenticated)
Pre-Condition	The user is logged in and has saved the team quota available.

Result	Team saved with tags, notes, and preview image.
Main Scenario	1. User clicks 'Save Team'. 2. The system stores data. 3. Confirmation displayed.
Alternative Scenarios	A1: Upload too large → show limit. A2: Network fail → allow retry.
Non-Functional Constraints	Upload limit 5MB. Save latency ≤ 1s. Privacy and backup enforced.

4.2.6. Use Case 6

Use Case ID	U006
Use Case	Create Discussion Post (Images + Caption)
Brief Description	An authenticated user creates a discussion post with images and text.
Actor	End User (authenticated), Database Administrator
Pre-Condition	Users authenticated and not banned.
Result	Post published or queued for moderation.
Main Scenario	1. The user uploads an image and writes a caption. 2. System validates and publishes.

Alternative Scenarios	A1: Flagged content → sent for moderation. A2: Upload invalid → re-upload prompt.
Non-Functional Constraints	Virus scanning enabled. Publish latency ≤ 2s.

4.2.7. Use Case 7

Use Case ID	U007
Use Case	React & Comment on Posts
Brief Description	Users interact with posts by adding reactions or comments.
Actor	End User (authenticated), Database Administrator
Pre-Condition	Target post available and unlocked.
Result	Reaction/comment saved and shown.
Main Scenario	1. User clicks react/comment. 2. System saves and updates counters.
Alternative Scenarios	A1: Post locked → disallow action. A2: Network fail → retry.

Non-Functional Constraints	Optimistic UI update. Rate-limited actions.

4.2.8. *Use Case 8*

Use Case ID	U008
Use Case	Account Registration & Login
Brief Description	The user registers an account, logs in, and manages authentication.
Actor	End User, Database Administrator
Pre-Condition	Authentication service operational.
Result	User logged in or account created successfully.
Main Scenario	1. Users sign up or log in. 2. System validates and issues sessions.
Alternative Scenarios	A1: Email exists → prompt reset. A2: Weak password → reject.
Non-Functional Constraints	Passwords hashed securely (bcrypt/argon2). HTTPS required.

4.2.9. *Use Case 9*

Use Case ID	U009
Use Case	View Post History

Brief Description	User can view posts and history profile of other users
Actor	End User (authenticated), Database Administrator
Pre-Condition	Accounts exist and are allowed.
Result	Followed user's posts visible in feed.
Main Scenario	1. User clicks follow. 2. System updates follow lists.
Alternative Scenarios	A1: Private account → pending approval. A2: Self-follow → blocked.
Non-Functional Constraints	Latency ≤ 500ms. Scalable follow system.

4.2.10. Use Case 10

Use Case ID	U010
Use Case	Customize Personal Information (Edit Profile)
Brief Description	User edits profile details such as name, password, or bio.
Actor	End User (authenticated), Database Administrator

Pre-Condition	User authenticated and authorized.
Result	Profile updated successfully.
Main Scenario	1. The user opens settings. 2. System validates and updates info.
Alternative Scenarios	A1: Weak password → reject. A2: Username conflict → show options.
Non-Functional Constraints	Strong validation and audit logs.

4.2.11. Use Case 11

Use Case ID	U011
Use Case	Moderate User-Submitted Content (Admin)
Brief Description	Admin reviews flagged content and moderates posts.
Actor	Admin/Moderator
Pre-Condition	Admin logged in with privileges.
Result	Content moderated, users notified.
Main Scenario	1. Admin reviews queue. 2. Takes moderation action.
Alternative Scenarios	A1: Legal escalation → escalate. A2: Undo → rollback action.

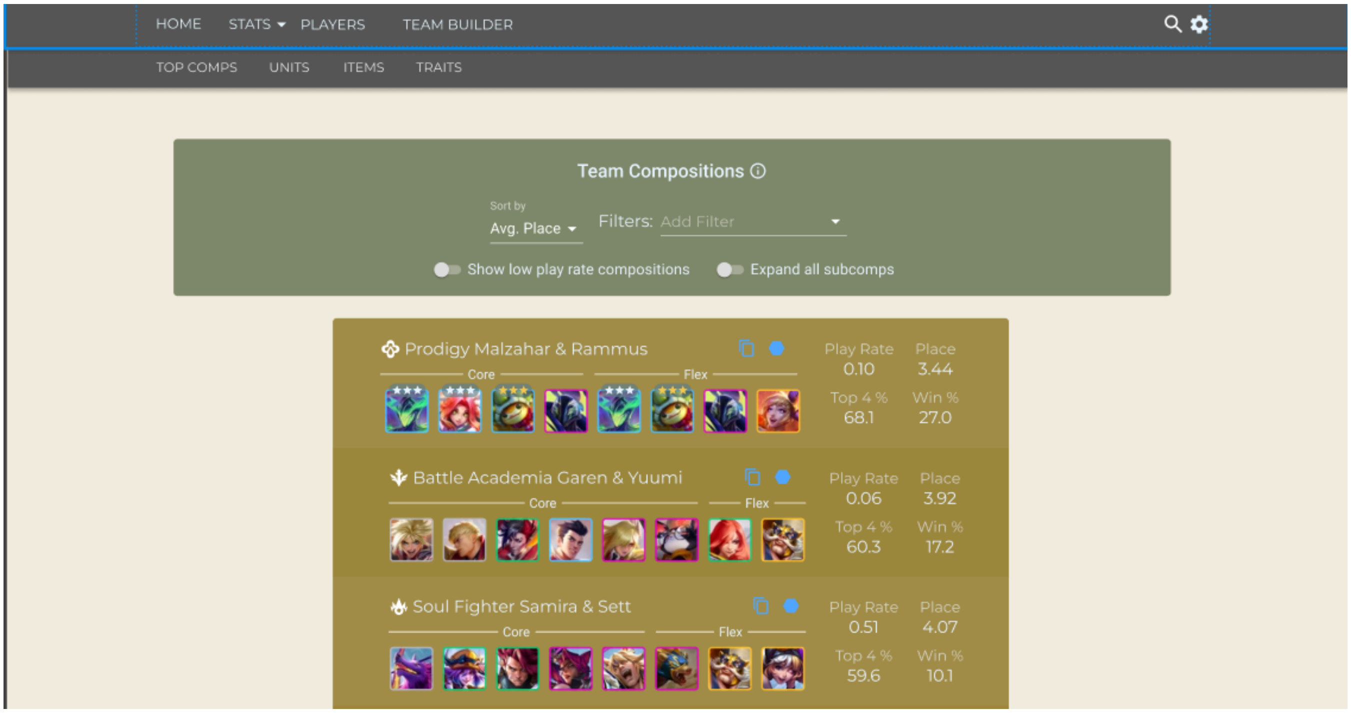
Non-Functional Constraints	Audit trail and role-based control.

4.2.12. Use Case 12

Use Case ID	U013
Use Case	Admin User Management (Lock/Unlock Accounts)
Brief Description	Admin locks or unlocks user accounts.
Actor	Admin, Database Administrator
Pre-Condition	Admin logged in with manage rights.
Result	User account status updated.
Main Scenario	1. Admin searches for users. 2. Executes lock/unlock.
Alternative Scenarios	A1: Already locked → show message. A2: Bulk lock → confirm action.
Non-Functional Constraints	Immediate session invalidation. Audit logging.

5 Prototype/Mockup

- **Figma:** Figma

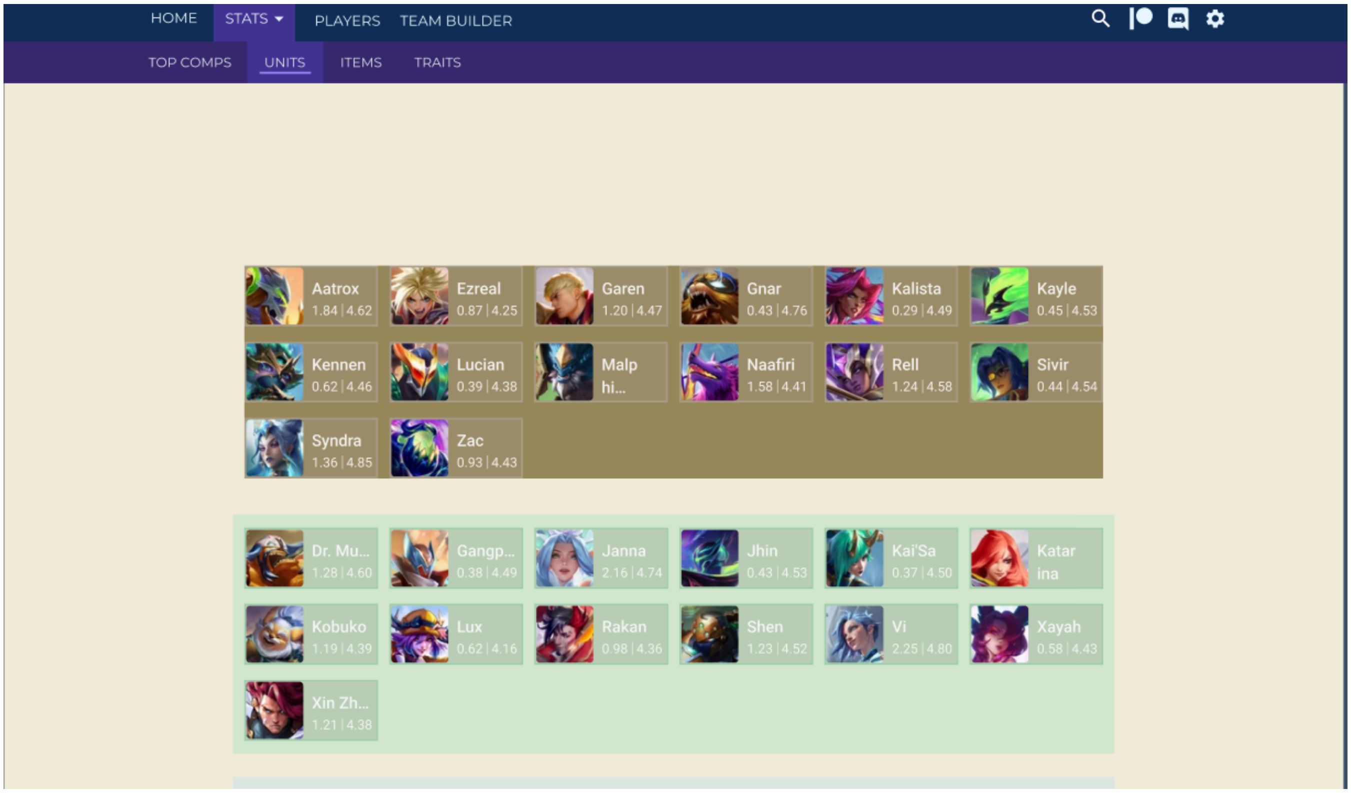


Prodigy Malzahar & Rammus — Play Rate 0.10, Place 3.44, Top 4 % 68.1, Win % 27.0

Battle Academia Garen & Yuumi — Play Rate 0.06, Place 3.92, Top 4 % 60.3, Win % 17.2

Soul Fighter Samira & Sett — Play Rate 0.51, Place 4.07, Top 4 % 59.6, Win % 10.1

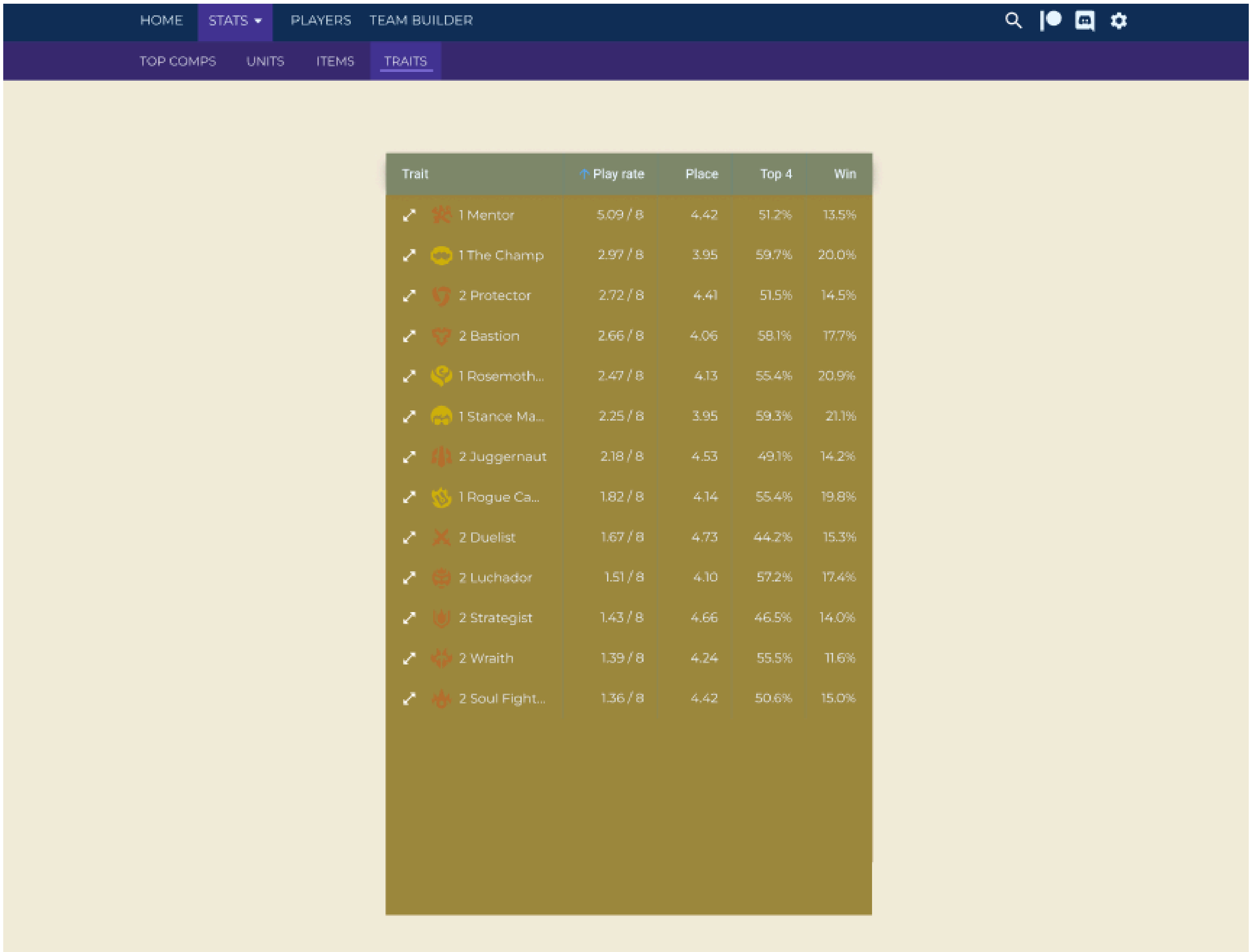
Home page



Aatrox 1.84 | 4.62	Ezreal 0.87 | 4.25	Garen 1.20 | 4.47	Gnar 0.43 | 4.76	Kalista 0.29 | 4.49	Kayle 0.45 | 4.53
Kennen 0.62 | 4.46	Lucian 0.39 | 4.38	Malp hi...	Naafiri 1.58 | 4.41	Rell 1.24 | 4.58	Sivir 0.44 | 4.54
Syndra 1.36 | 4.85	Zac 0.93 | 4.43				

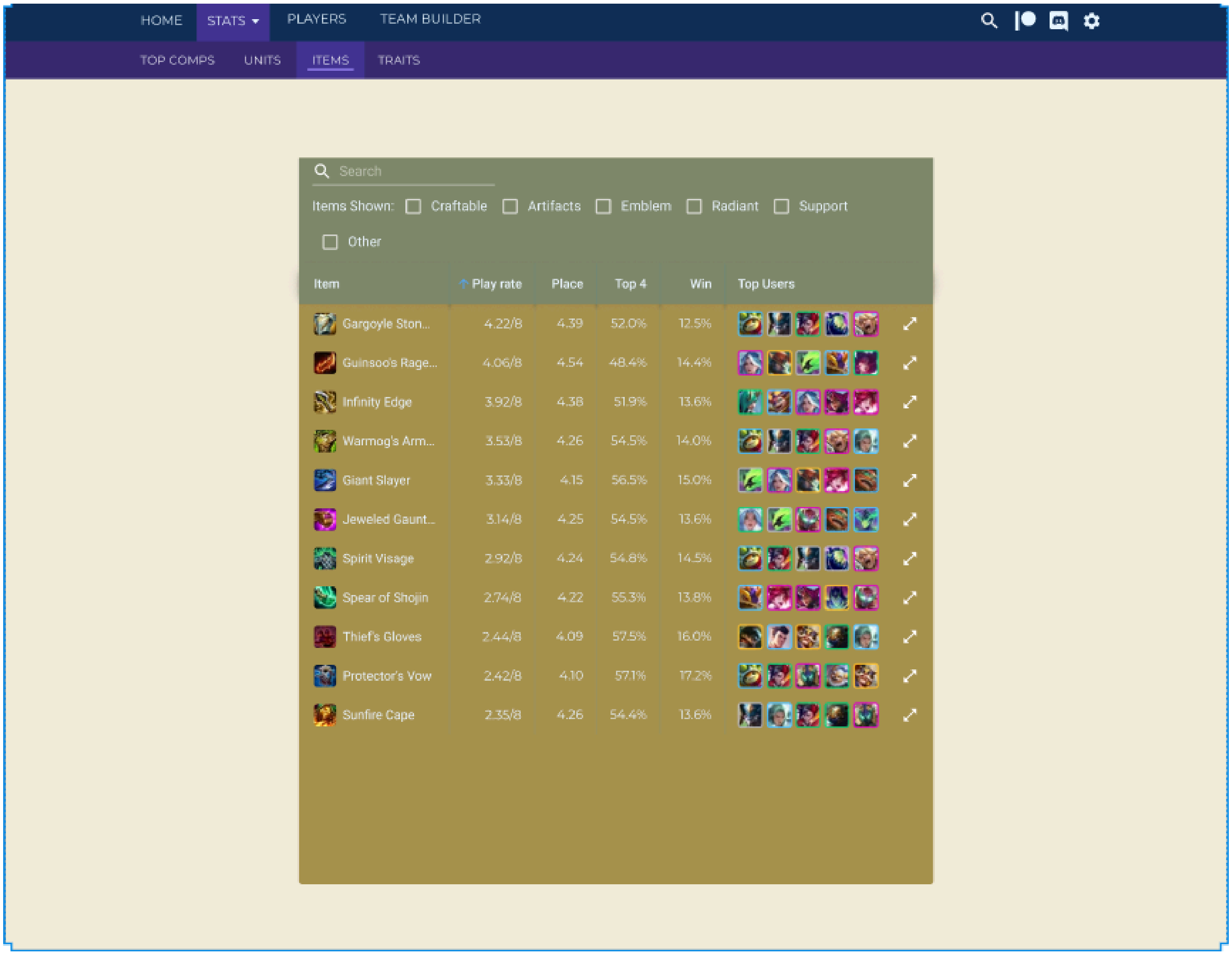
Dr. Mu... 1.28 | 4.60	Gangp... 0.38 | 4.49	Janna 2.16 | 4.74	Jhin 0.43 | 4.53	Kai'Sa 0.37 | 4.50	Katar ina
Kobuko 1.19 | 4.39	Lux 0.62 | 4.16	Rakan 0.98 | 4.36	Shen 1.23 | 4.52	Vi 2.25 | 4.80	Xayah 0.58 | 4.43
Xin Zh... 1.21 | 4.38					

Unit Page



Trait	↑ Play rate	Place	Top 4	Win
⤢ 1 Mentor	5.09 / 8	4.42	51.2%	13.5%
⤢ 1 The Champ	2.97 / 8	3.95	59.7%	20.0%
⤢ 2 Protector	2.72 / 8	4.41	51.5%	14.5%
⤢ 2 Bastion	2.66 / 8	4.06	58.1%	17.7%
⤢ 1 Rosemoth...	2.47 / 8	4.13	55.4%	20.9%
⤢ 1 Stance Ma...	2.25 / 8	3.95	59.3%	21.1%
⤢ 2 Juggernaut	2.18 / 8	4.53	49.1%	14.2%
⤢ 1 Rogue Ca...	1.82 / 8	4.14	55.4%	19.8%
⤢ 2 Duelist	1.67 / 8	4.73	44.2%	15.3%
⤢ 2 Luchador	1.51 / 8	4.10	57.2%	17.4%
⤢ 2 Strategist	1.43 / 8	4.66	46.5%	14.0%
⤢ 2 Wraith	1.39 / 8	4.24	55.5%	11.6%
⤢ 2 Soul Fight...	1.36 / 8	4.42	50.6%	15.0%

Traits Page

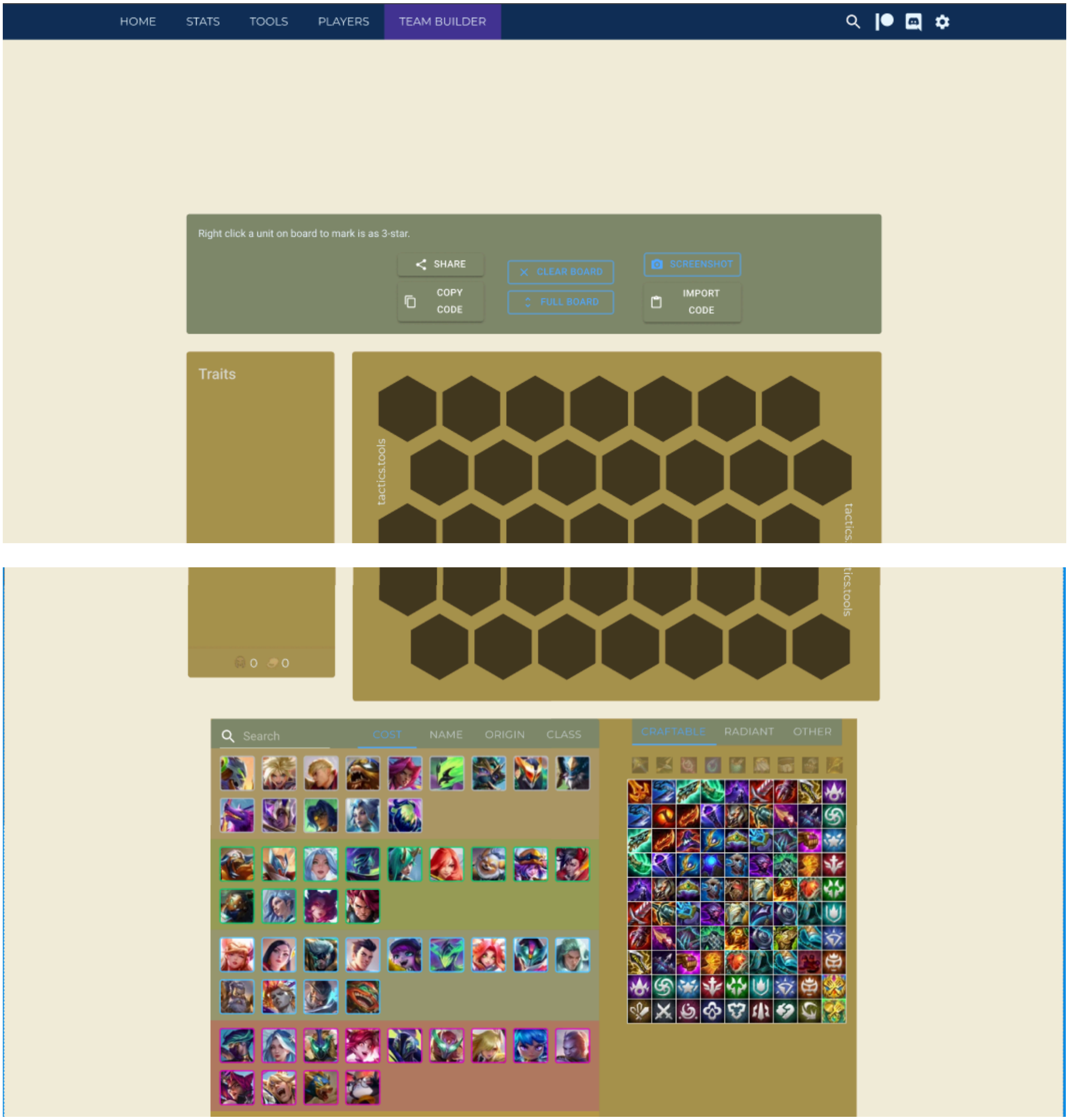


Items Shown: □ Craftable □ Artifacts □ Emblem □ Radiant □ Support

□ Other

Item	↑ Play rate	Place	Top 4	Win	Top Users
Gargoyle Ston...	4.22/8	4.39	52.0%	12.5%	
Guinsoo's Rage...	4.06/8	4.54	48.4%	14.4%	
Infinity Edge	3.92/8	4.38	51.9%	13.6%	
Warmog's Arm...	3.53/8	4.26	54.5%	14.0%	
Giant Slayer	3.33/8	4.15	56.5%	15.0%	
Jeweled Gaunt...	3.14/8	4.25	54.5%	13.6%	
Spirit Visage	2.92/8	4.24	54.8%	14.5%	
Spear of Shojin	2.74/8	4.22	55.3%	13.8%	
Thief's Gloves	2.44/8	4.09	57.5%	16.0%	
Protector's Vow	2.42/8	4.10	57.1%	17.2%	
Sunfire Cape	2.35/8	4.26	54.4%	13.6%	

Item Page



Right click a unit on board to mark is as 3-star.

Team Builder