



League of Legend Analytics

Group-23

Made By:

Smit kumbhani - 201801404

Bhargav Dave - 201801402

Shreeya Godbole - 201801263



Purpose of Project

- League of Legends a.k.a. LoL is an extremely famous MOBA game.
- The sheer number of heroes, items and strategies makes playing it extremely difficult.
- There are too many possibilities and this can be overwhelming for new players because of the amount of combinations possible.
- There is no one tool which provides a detailed repository of the game mechanics as well as a tool that shows descriptions of various heroes and items and can in fact show the various suggested combinations and strategies, ones that are computer recommended as well as ones that are used by the top players of the world.
- We are setting out to create such a tool for players, old and new, of LoL so that they get the resources they need in an elegant fashion.



Scope of project

- We are aiming to show details like win rates, hero win rates, leaderboards, comparison charts, as well as a log of recent performances in order for a player to better analyze and improve their playstyle and gameplay.
- We are aiming to add a functionality which allows the players to log in and check on the strategies that might be recommended to them and tailor made for their profiles.

Deliverables:

- Create a website which has various functionalities which include: User sign-up, User login, Password change, Tier List, Real-time analysis, Comparison, Personal stats.
- Create a holistic software that contains tools helpful for the old and new players of LoL.



Who will use the system?

- Gamers

People who are playing the games to find out which character is best for them and which attack - defense combination can lead them to victory.

- Admin

People whose work is to improve, remove or add new functionalities in the system and for maintenance purpose.



Who can provide input about the system?

The following can provide creative inputs about the system:

- Investors: The people who are investing in the project will provide input as they are financing the project and will get affected by its outcomes.
- Users/Gamers: They can give suggestions and feedback about the software , this will help improve the project.
- Developers: They are the team members who can provide their thoughts and suggestions to better the project.

The following can provide data for our system:

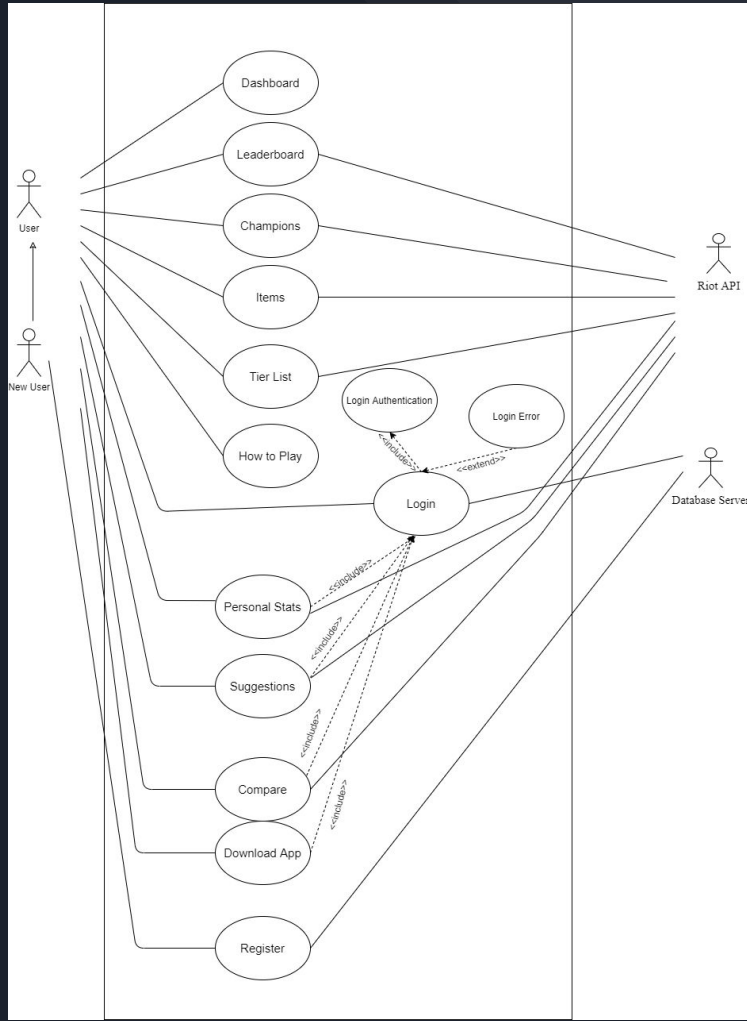
- API: It will fetch data from the LoL database and provide raw data for our system on which we can perform analysis and generate reports and results.



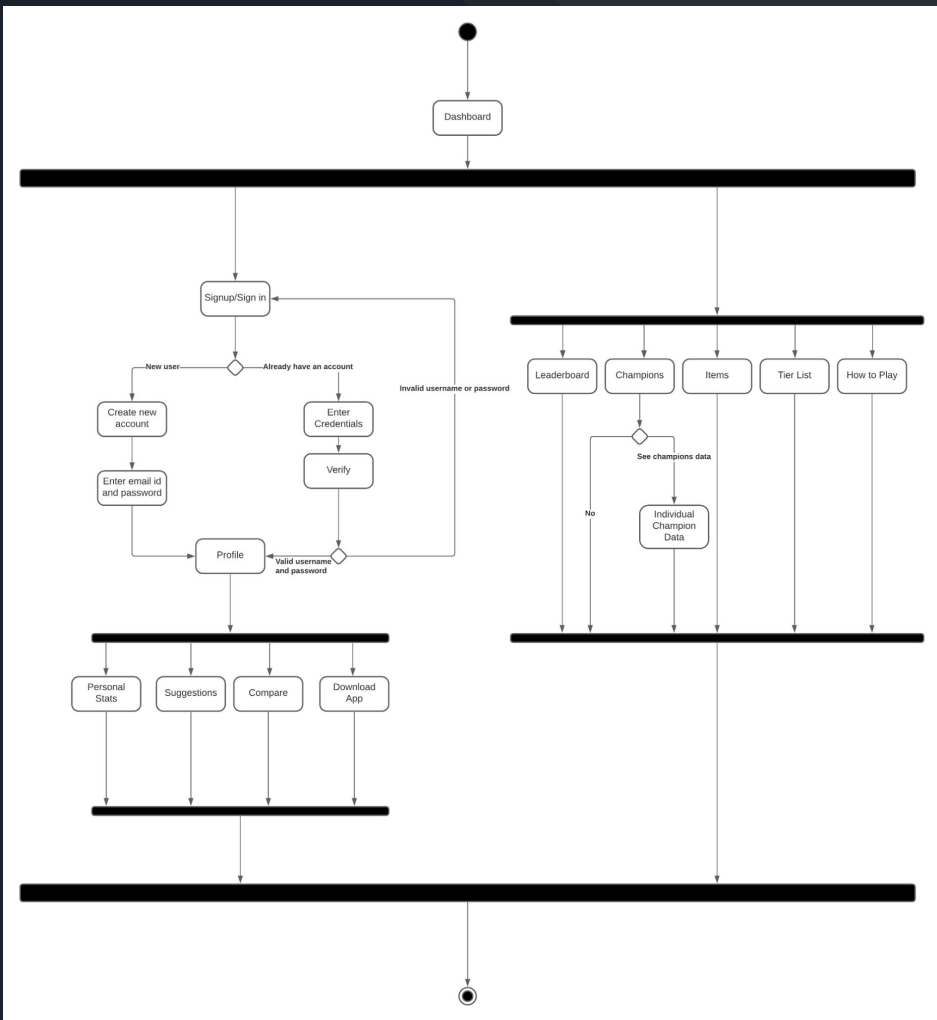
Business rules for our system:

- All users are required to have a valid email address and password to access the system.
- Users data namely email address should be stored securely and will not be used by admin for any purpose.
- Data which is used by the system to calculate the stats should be up to date with the source.

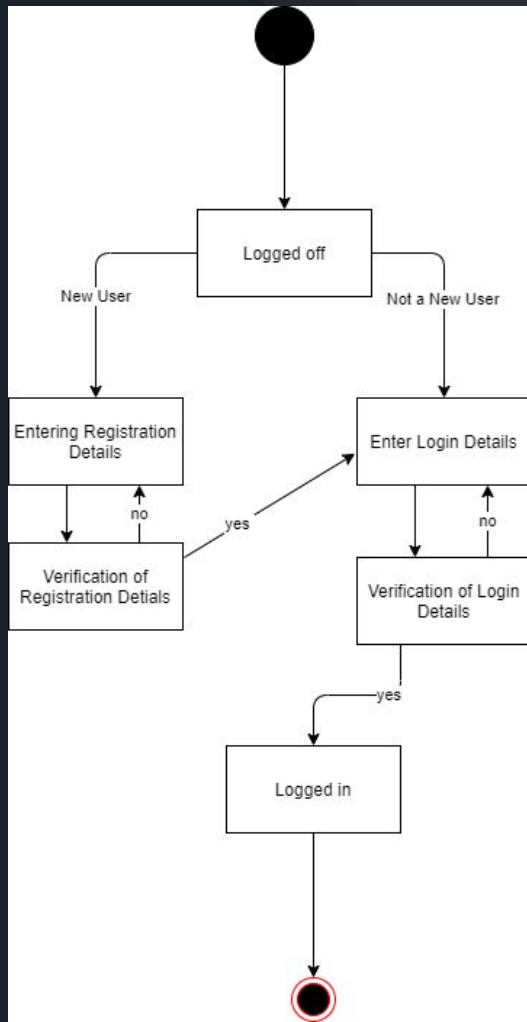
Use Case Diagram



ACTIVITY DIAGRAM

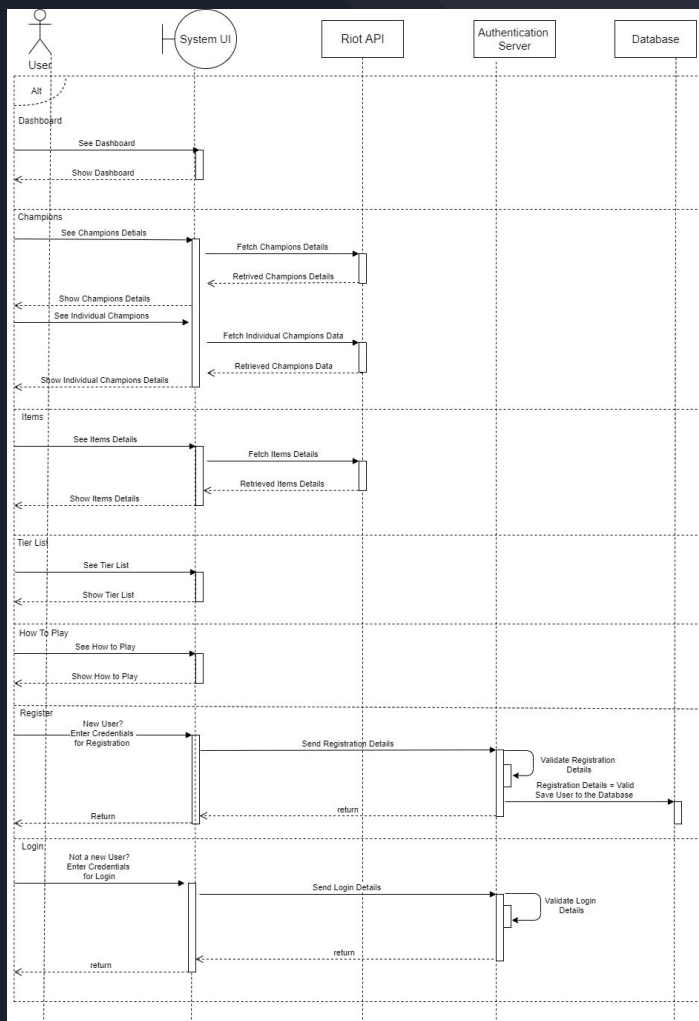


STATE DIAGRAM



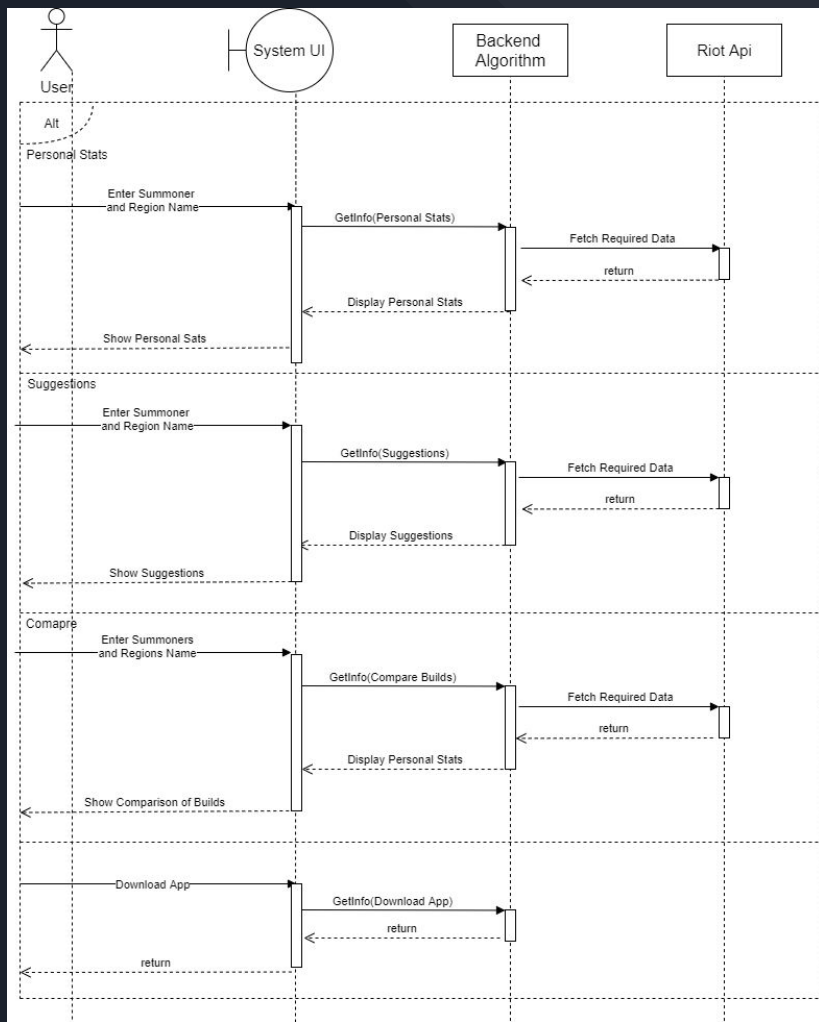
SEQUENCE DIAGRAM

Before Login



SEQUENCE DIAGRAM

After Login





Task Dependencies

A. Requirements:

1. Gather requirements
2. User stories (1)
3. Write functional ,
non-functional
requirements (2)
4. Plan project flow (1)
5. Search API (4)

B. Design

6. Create timeline (4)
7. Concept map (1)
8. Activity Diagram (4)
9. Use-case diagram
(2, 4)
10. State diagram (4)
11. Sequence diagram
(10)

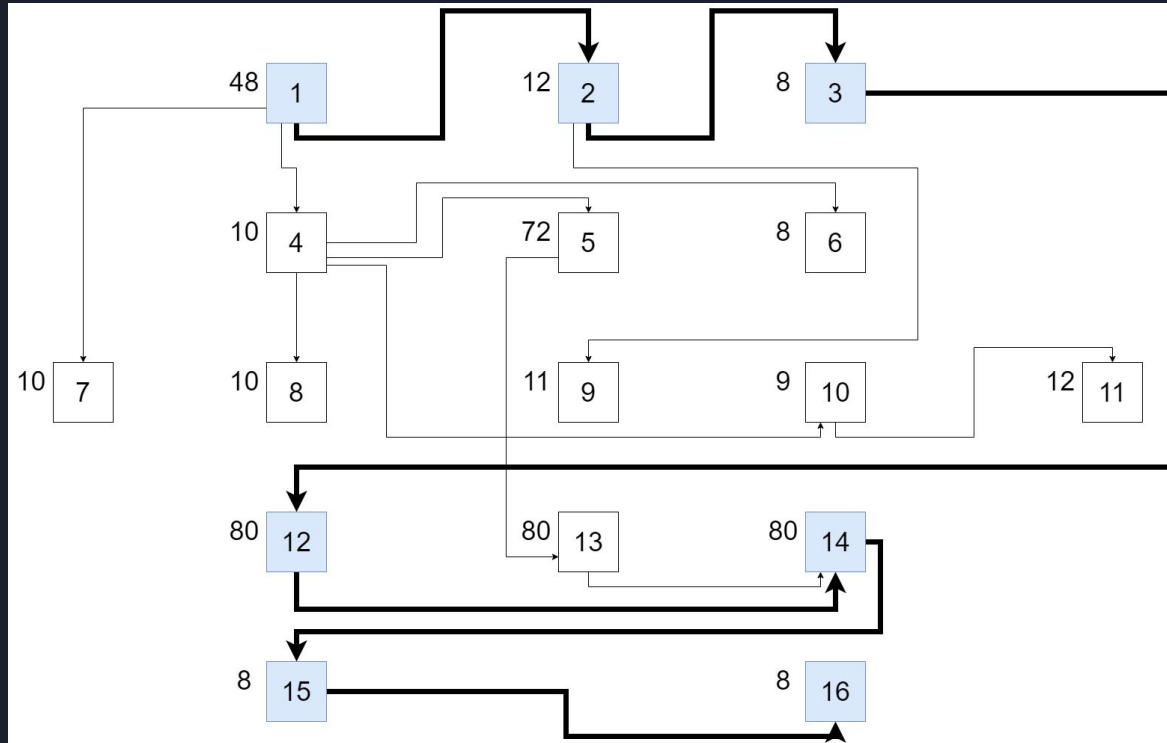
C. Implementation

12. Home page (3, 6)
13. Backend data
fetching (5)
14. Linking backend to
frontend (12, 13)

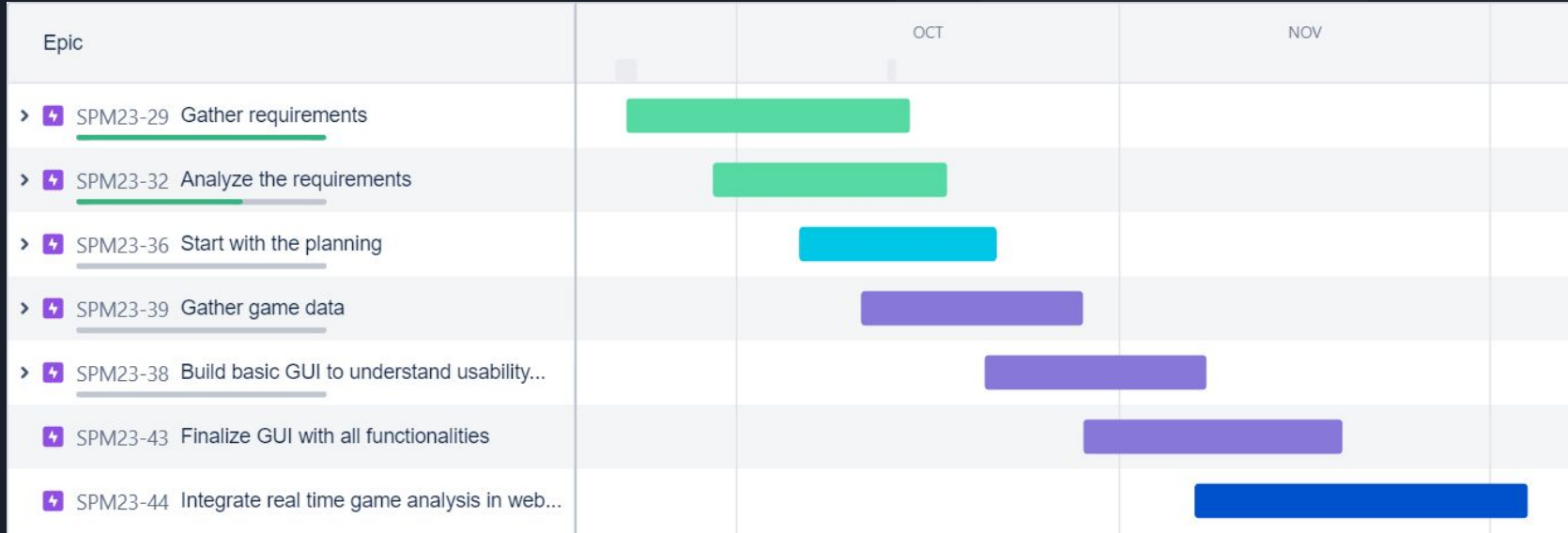
D. Testing

15. Black box testing
(14)
16. GUI testing (15)

Critical Path



Gantt Chart





Design and implementation constraints

- Limitation of Data
- Limited API calls
- Constraint on Scalability
- Lack of resource to keep updating Database
- Incapability of SQLite with version control software



Risks and Mitigation

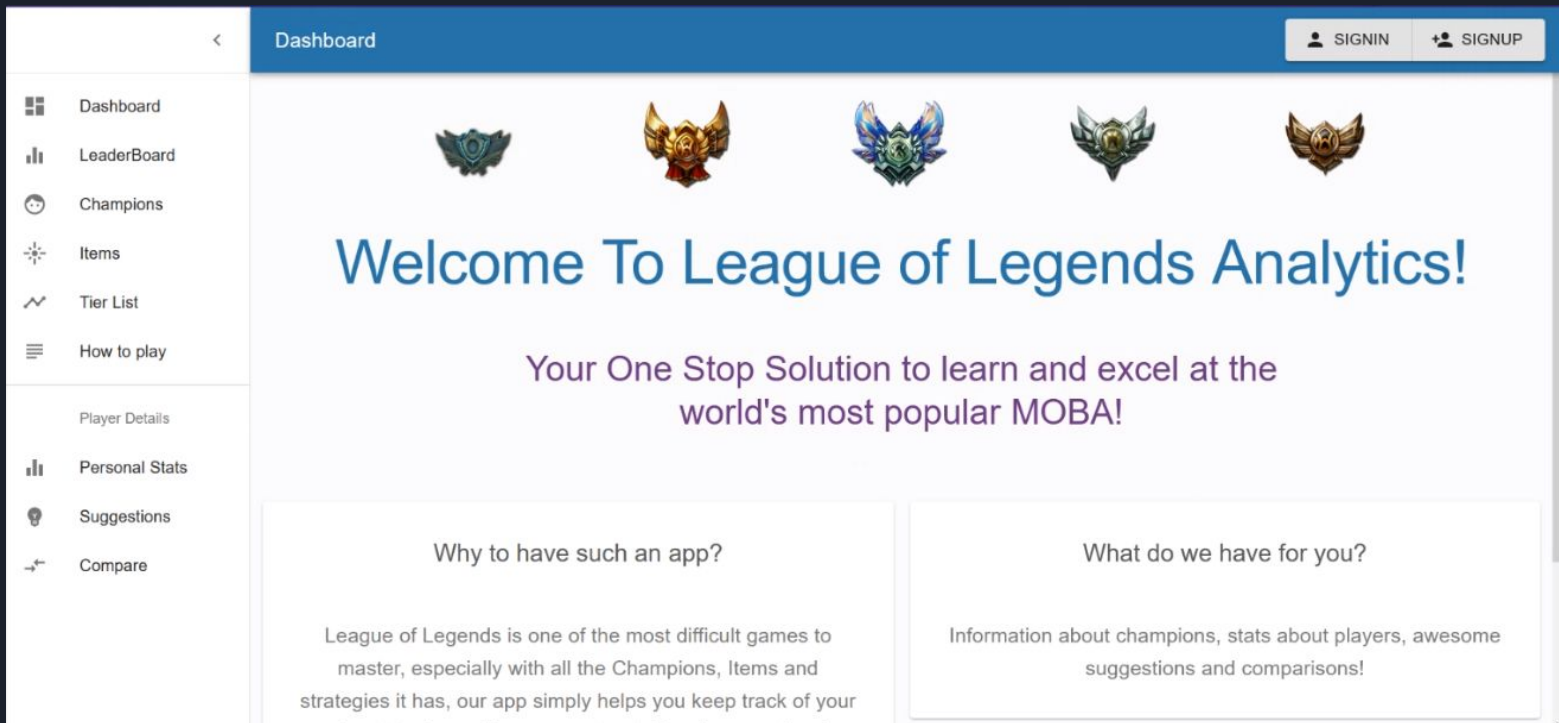
Risks:

- Free API, limited calls
- Local Machine, system can't withstand more user traffic
- No use of authentication for login
- No data encryption, may lead to data leakage

Mitigation:

- Upgrade to a paid version
- Deploy on a proper server
- Integrating authentication measures
- Use a proper encryption algorithm for data privacy

UI/UX design of the complete system



UI/UX

<

LeaderBoard

SIGNIN

SIGNUP

Dashboard

LeaderBoard

Champions

Items

Tier List

How to play

Player Details

Personal Stats

Suggestions

Compare

League

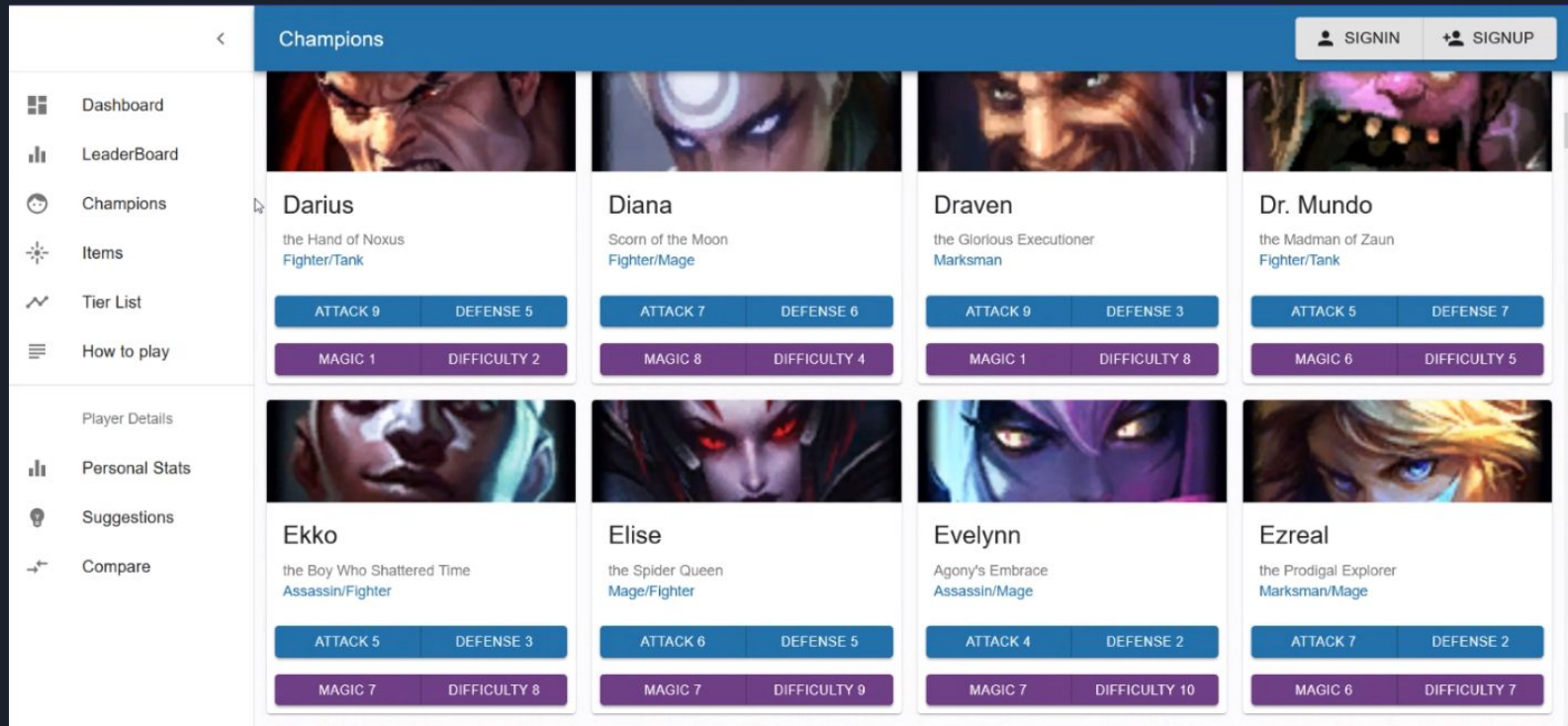
Queue Type

Region

SUBMIT

Rank	Summoner Name	LP	Wins	Loss	Win Rate
1	EDG ME2	883	81	59	57.86 %
2	Buu	875	177	151	53.96 %
3	Worthless	875	109	98	52.66 %
4	Insightful	875	166	115	59.07 %
5	Lewiak	875	96	44	68.57 %
6	FBI OPEN BOTLANE	874	257	229	52.88 %
7	Szafer	874	223	186	54.52 %
8	Rinkhalz	874	186	136	57.76 %
9	A1M	873	409	373	52.30 %

UI/UX



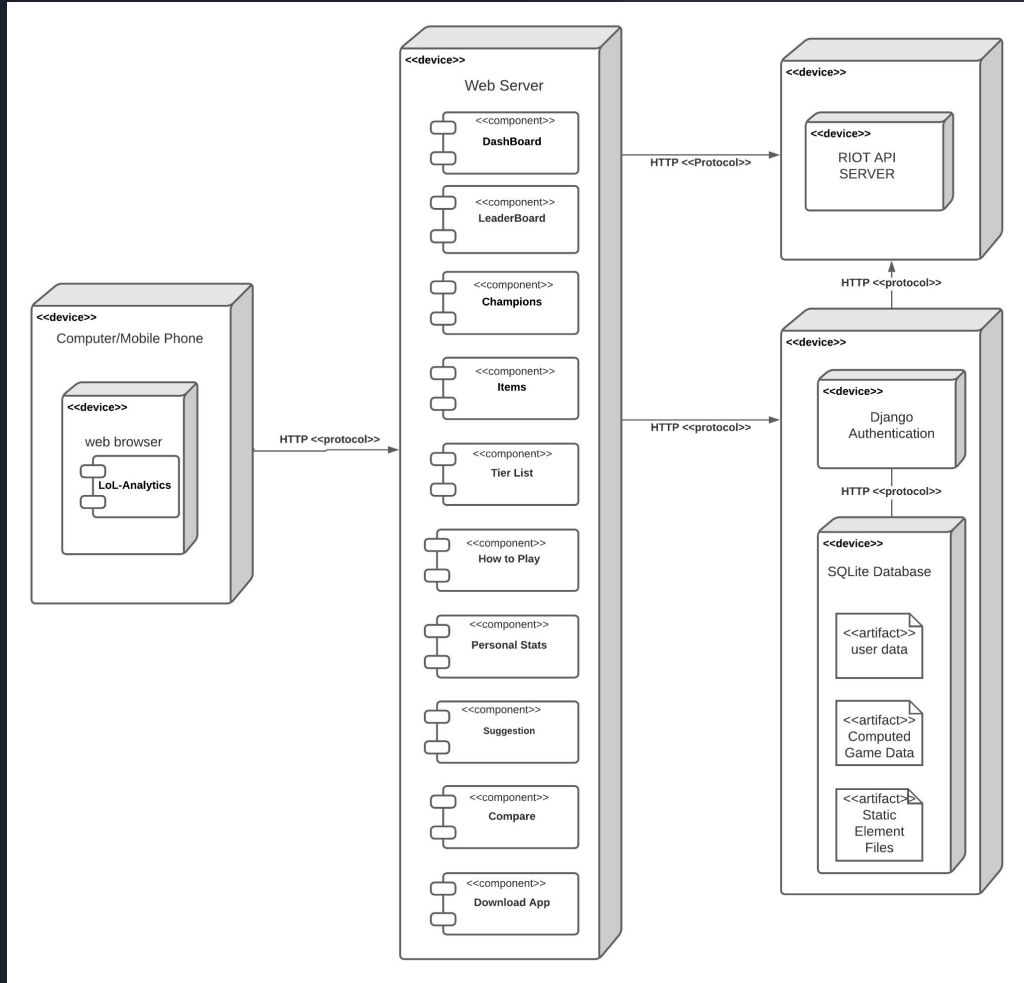
- Dashboard
- LeaderBoard
- Champions
- Items**
- Tier List
- How to play
- Player Details
- Personal Stats
- Suggestions
- Compare

Items

SIGNIN SIGNUP

		BUY 450	SELL 315
<p>Long Sword Damage/Lane</p> <p>SLIGHTLY INCREASES ATTACK DAMAGE</p> <p>BUY 350 SELL 245</p>	<p>Pickaxe Damage</p> <p>Moderately increases attack damage</p> <p>BUY 875 SELL 613</p>	<p>B. F. Sword Damage</p> <p>GREATLY INCREASES ATTACK DAMAGE</p> <p>BUY 1300 SELL 910</p>	<p>Hailblade LifeSteal/SpellVamp/Jungle</p> <p>PROVIDES DAMAGE AGAINST MONSTERS AND MANA REGEN IN THE JUNGLE</p> <p>BUY 350 SELL 140</p>
<p>Dagger AttackSpeed</p> <p>Slightly increases attack</p>	<p>Recurve Bow AttackSpeed/OnHit</p> <p>GREATLY INCREASES ATTACK</p>	<p>Amplifying Tome SpellDamage</p> <p>Slightly increases ability</p>	<p>Vampiric Scepter Damage/LifeSteal</p> <p>BASIC ATTACKS RESTORE HEALTH</p>

System Architectural model





Implementation Technologies

Frontend:

- react.js
- node.js
- Material UI

Backend:

- Python 3
- Django
- SQL



Contribution

Smit Kumbhani	User stories, Requirement gathering, Planning workflow, Making concept map, Making state diagram, Implementing home screen on react, Backend code on python, Linking of backend and frontend
Bhargav Dave	Functional and Non-functional requirements, User stories, Project Planning, Finding API, Activity diagram, Sequence diagram, Implementing home screen on react, Backend code on python, Linking of backend and frontend, GUI testing
Shreeya Godbole	Gathering requirements, Writing user stories, Project planning, Creating Timelines, Use case diagram, Implementing home screen on react, Backend code on python, Linking of backend and frontend



Future Enhancements

- Program for renewing API key or a paid API
- Deployment on a server for tackling more user traffic
- Use of encryption for data privacy and use authentication for security measures.



Open/unresolved issues

- Could not implement forgot password functionality
- API key expires every 24 hours so key needs to be refreshed frequently
- A few users' personal statistics are not available publicly so their data is not fetched by the API.

Thank you!

