Betting System

Target release	End Of Trial Period
Document status	DRAFT
Document owner	@kim @Steve Ops @Brian Wachira @Kevin Ledama
Designer	@ designer
Tech lead	@kim
QA	

🎯 Objective

The purpose for this document is to define requirements that allow a software engineer to build a minimum viable product for a betting system.

The system should be able to do the following -

- A user should be able to register, then login
- Users may have up to 2 levels of access;
  - Frontend Access - manage your own account
  - Admin Access - manage accounts of any user
- By default, a frontend user should be able to;
  - View sport games. Limit this to football only but system should be dynamic enough to accommodate other types
  - Place bets on these games - assume they have unlimited money in their wallets
  - Cancel bets on these games
  - View history of their bets
  - View accounts of winnings and losses
- By default, an admin user should be able to;
  - View a user, with all games they have placed
  - Soft delete a user with all associated data
  - View profits made from game losses
- An admin, with `superuser` permission, should be able to;
  - Configure sport games
  - Grant admin access to a user
  - Revoke admin access to a user
- Send an email when a bet is won/lost

Note: **READ ABOUT GENERATORS BUT LIMIT THE USE OF GENERATORS** in creating the relevant modules

📊 Success metrics

Goal	**Metric**

Registration	Individuals should be able to self register
Sign-In	Individuals should be able to login
Roles and Permission	An admin should be able to; • Add another admin • Revoke an existing admin • Make an existing admin a `superuser` • Revoke `superuser` right • Add sport games • View profits made from game losses

🤔 Assumptions

Assume that the client for this system requires users to register with the following details:-

- **first_name**
- **last_name**
- **email_address**
- **msisdn**

You can add as many columns as necessary to present a holistic system.

Please feel free to have unit tests within the implementation.

In addition, demonstrate the use of **priority queueing** mechanism using any available **elixir** binding.

If using a relational database, demonstrate a deep understanding of relations, indexes, constraints within the different schemas.

❓ Open Questions

Question	Answer
When is the expected review to happen ?	Solution will be reviewed **2 days** before end of trial period.