Betgogo23

Technical Document

UNIVERSITY OF BIRMINGHAM

JUNIOR SOLUTION

A project presented by Birmingham Innovation Studio

Date: 2020-12-17

1. Introduction

This document explains the functional specification of safequeen, the web app project in which Birmingham Innovation Studio, engages to fulfill all the requests demanded by safequeen as explained during elementary meetings and calls.

1.1 Purpose of this document

This document delineates the functional and non-functional requirements identified and proposed by Innovation Studios for the project under consideration. The functional requirements capture the essential functionalities and behaviors the system or solution is expected to achieve, while the non-functional requirements detail the quality attributes, performance standards, and other supplementary characteristics. Our objective is to provide a comprehensive, unambiguous, and structured outline that ensures both parties have a shared understanding of the project's expectations and deliverables. This document has been meticulously crafted by bernard, in alignment with the requirements and expectations of safequeen. We encourage our clients to review this document meticulously and engage with us for any clarifications or further discussions.

1.2 Scope

betgogo is a web app that allows user to place bet between each other, this app will have a login and sign up page where users can login or register, reset their password and reset their email. It will have a profile page where users can update their profile and manage their account. there will be a friend page where users can mlanage their relationship with others. there will be a bet page where users can see their current bet, create new bets and modify existing bets.

1.3 References

All meetings, emails and materials exchanges with safequeen.

2 Requirements

2.1 Functional Requirements

1. Authentication Pages:

- The web app must allow users to sign in and sign up using email and password.
- The web app must include a 'Forgot Password' feature that sends a reset link to the user's email.
- The web app should offer social media login options like Facebook and Google.
- Success message must be displayed after successful registration, login, or password reset.

2. Profile Page:

- The web app must offer a profile page where users can update their personal information.
- The web app should allow users to upload a profile picture.
- The web app must provide options for users to manage their account settings.
- Success message must be displayed after successfully updating the profile or account settings.

3. Friends Page:

- The web app must offer a Friends page where users can manage their relationships.
- The web app should allow users to send and accept friend requests.
- The web app must provide options for blocking or unfriending users.

• Success message must be displayed after successfully sending, accepting, or declining a friend request.

4. Bet Page:

• The web app must offer a Bet page where users can manage their current bets.

• The web app should allow users to create new bets.

• The web app must provide options for modifying or canceling existing bets.

• Success message must be displayed after successfully creating, modifying, or canceling a bet.

2.2 Non-Functional Requirements

1. Reliability:

• The web app must have a minimum uptime of 99.99%.

• The web app should have a failover mechanism to switch to a backup service in case of server failure.

• The web app should offer offline access to user profiles and account settings through caching.

2. Performance:

• The web app should load user data and bet information within 3 seconds.

• The web app should be optimized for low-latency data fetching for real-time bet updates.

• The web app should be able to handle a minimum of 1000 concurrent users.

3. Usability:

- The web app must be user-friendly and offer a tutorial for first-time users.
- The web app should be accessible, following WCAG 2.1 guidelines for web apps.
- The web app should be responsive and compatible with different devices and screen sizes.

4. Security:

- All user data must be encrypted both in transit and at rest using AES-256 encryption.
- The web app must comply with data protection regulations like GDPR.
- The web app should have a secure authentication mechanism, including two-factor authentication.
- Access to user data should be logged and regularly audited for security breaches.

5. Scalability:

- The web app should be designed to handle an increasing number of users and data sets.
- The web app should be cloud-native to allow for seamless scaling.
- The web app should be modular to allow for the addition of new features without affecting existing functionality.
- The web app's database should be designed for horizontal scaling to accommodate growing data.