Betgogo23

Technical Document

UNIVERSITY OF BIRMINGHAM

JUNIOR SOLUTION

A project presented by Birmingham Innovation Studio

Date: 12/01/2002

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

The functional specification document serves the same purpose as a contract. The specifications presented in this document are related to the conversations done by bernard. Following this document, the developers agree to provide the capabilities specified and fulfill all requests and demands of safequeen, who also agreed to find the product satisfactory if it provides the capabilities as specified.

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 using email and password.
- The web app must allow users to register using their email and create a password.
- The web app must offer a 'Forgot Password' feature that sends a reset link to the user's email.
- The web app should provide an option for two-factor authentication.
- Success message must be displayed after successfully registering, logging in, or resetting password.

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 display the user's betting history.
- Success message must be displayed after successfully updating the profile.

3. Bet Page:

- The web app must display the user's current bets, including opponents and wager amount.

• The web app should allow users to create new bets by specifying opponents and wager amount.

• The web app must allow users to modify or delete existing bets.

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

4. Friends Page:

• The web app must offer a friend page where users can manage their relationships.

• The web app should allow users to send friend requests and accept or decline requests.

• The web app must display the user's current friends list.

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

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 betting history.

2. Performance:

• The web app should load profile and betting information within 2 seconds.

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

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.

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 such as GDPR.
- The web app should have a secure authentication mechanism, including two-factor authentication.

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 easy 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.