

Bier Pongo

Kevin Fischer
OTH Amberg-Weiden
k.fischer2@oth-aw.de

Markus Fleischmann
OTH Amberg-Weiden
m.fleischmann2@oth-aw.de

Darren Fürst
OTH Amberg-Weiden
d.fuerst@oth-aw.de

Simon Kleber
OTH Amberg-Weiden
s.kleber1@oth-aw.de

Fabian Klinger
OTH Amberg-Weiden
f.klinger1@oth-aw.de

Jakob Lindner
OTH Amberg-Weiden
j.lindner3@oth-aw.de

Christian Renner
OTH Amberg-Weiden
c.renner@oth-aw.de

Abstrakt— Cloudfähige, skalierbare, multiuser- und multidevice-fähige Bier Pong Social Media Plattform.

Schlüsselwörter—Beer Pong, Cloud, Data, Team

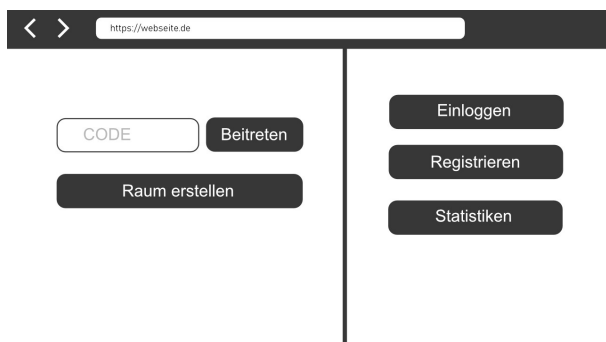
I. EINLEITUNG

Bei Bier Pongo handelt es sich um eine Webanwendung, die es den Benutzern ermöglicht ihr reelles Beer Pong Spiel im Web zu hosten. Beer Pong (auch Beirut oder Bier Pong) ist ein Trink- und Geschicklichkeitsspiel sowie ein Präzisionssport, bei dem Tischtennisbälle in Becher geworfen werden müssen.^[1] In dem Spiel stehen sich Teams, nahezu beliebiger Größe, an einer Bierbankgarnitur, der Länge nach, gegenüber. Die Becher, die vom gegnerischen Team getroffen werden müssen, stehen in Form einer Pyramide, vor dem jeweiligen Team. Vor Spielbeginn, wird einheitlich beschlossen mit wie vielen Becher pro gespielt wird, die Spieler treffen ihre Auswahl zwischen 6, 10 oder 15 Bechern. Welches Team beginnt wird durch einen fairen Zufallsprozess ausgelost. Pro Runde erhält jedes Team zwei Tischtennisbälle, wurden beide Bälle verworfen, ist das andere Team an der Reihe. Wird ein Becher vom gegnerischen Team getroffen, so muss sich das Besizerteam der Becher den Inhalt (i.d.R. Bier) einverleiben, werden zwei Becher getroffen müssen auch diese getrunken werden und zudem erhält das Team die Bälle zurück und darf erneut

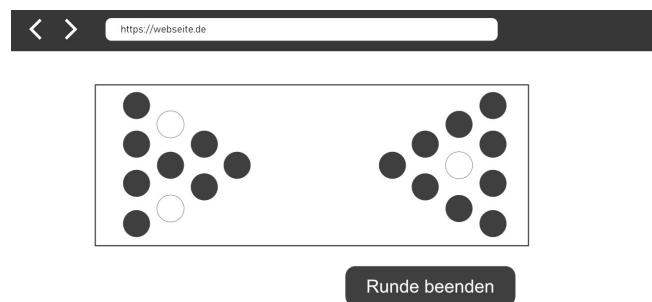
werfen. Das Team, dass es zuerst schafft, alle Becher des gegnerischen Teams zu treffen, gewinnt. Die Anwendung Bier Pongo schafft den Spielern die Möglichkeit, gegeneinander anzutreten, auch wenn Sie gerade keine Möglichkeit haben, sich persönlich gegenüberzustehen. Bier Pongo setzt Ehrlichkeit und das Befolgen der allgemein bekannten Regeln voraus.

A. Motivation

Als Motivation und Ideengeber galt für das Team die Corona-Krise. Für viele Jugendliche und Alkoholbegeisterte zählt das Beer Pong Spiel zum wesentlichen Zeitvertreib, an Wochenenden und Feiertagen. An Geburtstagen und anderen Feiern, ist Beer Pong als ein konkurrenzloser Prozess bekannt, um den Alkoholgehalt im Blut schnellstmöglich und mit Spaß auf die Spitze zu treiben. Durch Kontaktbeschränkungen und Verbote war es vielen Spielern nicht mehr möglich, dieses Spiel auszuüben. Unzählige waren davon schwer getroffen. Um solche Schicksalsschläge in Zukunft möglichst zu vermeiden, möchte das Team mit Bier Pongo allen Nutzern kostenlos die Möglichkeit geben, online via remote ihren



Entwurf Startbildschirm



Spielansicht

alkoholischen Gelüsten, mit Spiel und Spaß, nachzukommen.

B. Alleinstellungsmerkmal

Als erste remote spielbare Bier-Pong-Anwendung will Bier Pongo mit möglichst simplen und intuitiv bedienbaren Elementen die Spieler beim realen Spiel unterstützen. Auf aufwändige Animationen sowie Werbung wird komplett verzichtet.

Die Webanwendung ist als übersichtliches Infoboard gedacht, das parallel zum realen Spiel genutzt wird. Beispielsweise per Notebook auf einem Fernseher, damit wiederum per Videostream nicht das Spielfeld, sondern die Spieler im Fokus stehen können.

Je nach Projektfortschritt können sich besonders ambitionierte Spieler auch über Einsichten in verschiedene Statistiken sowie eine Möglichkeit zum User-Login freuen.

II. VERWANDTE ARBEITEN

Inspiration für das Projekt bot die Online-Funktion der Dart-App „Pro Darts“, die ähnlich wie Bier Pongo für das Remote-Spielen genutzt werden kann.^[2] Dabei können zwei räumlich getrennte Spieler ihre geworfenen Dart-Scores abwechselnd eintragen. Die Spielstände werden dann jeweils live aktualisiert.

Für Dartfreunde, die sich nicht privat treffen können ist das zu Zeiten von Lockdowns und auch unabhängig davon eine toller Möglichkeit, dennoch zusammen dem gemeinsamen Hobby nachzugehen.

Bier Pongo will dieses Prinzip des rundenbasierten Scorings für das Trinkspiel Bier-Pong adaptieren.

III. ANFORDERUNGEN

Die Anforderungen werden in Form von User-Stories mit Akzeptanzkriterien formuliert. Die MUSS-Anforderungen definieren das „Minimum Viable Product“ (MVP). Die SOLL-Anforderungen sollen in den nachfolgenden Iterationen fertiggestellt werden, wobei die Userstories unter OPTIONAL mögliche Erweiterungen für das Produkt bieten, welche je nach Zeitbedarf umgesetzt werden können.

A. MUSS-Anforderungen

1) Raum erstellen

Ich als Nutzer möchte über einen Knopf einen neuen Raum erstellen können, um dort ein neues Spiel zu starten, welches am Anfang auf „Wartend“ steht.

Akzeptanzkriterien:

- Weiterleitung in neuen Raum nach Button-Druck
- Anzeige der Raum-ID für diesen Raum
- Status des Raums wird als wartend angezeigt
- Volles Spielfeld wird angezeigt
- Keine Spielinteraktion möglich, da noch nicht vollständig

2x 20, 2x 13

2) Raum beitreten

Ich als Spieler möchte nach Erhalt eines Codes durch einen anderen Spieler nach Eingabe des Codes mittels

einem Knopfdrucks dessen Raum beitreten, um dort als Gegenspieler anzutreten.

Akzeptanzkriterien:

- Nach Eingabe des Codes wird bei einem ungültigen Code eine Meldung an den Nutzer ausgegeben.
- Bei einem gültigen Code wird man in den Raum weitergeleitet.
- Volles Spielfeld wird angezeigt. 4x 5
- Status des Raums wird auf Bereit gesetzt

3) Spiel eintragen

Ich als Spieler möchte mein Ergebnis eintragen können, um meinen Spielstand meinem Gegenspieler mitzuteilen.

Akzeptanzkriterien:

- Möglichkeit, dass Becher nur ausgewählt werden, wenn am Zug ist
- Becher dürfen nur ausgewählt werden können, wenn sie noch voll sind
- Nach Speichern des Zuges wird der Bildschirm aktualisiert und der neuen Zustand des Tisches wird dargestellt
- Der Gegenspieler erhält mit der nächsten Auffrischung das aktuelle Spielgeschehen
- Wird das Spiel durch einen Spieler gewonnen, so werden den Spieler eine “Gewonnen/Verloren” Nachricht angezeigt, mit einem nun sichtbaren Button “Zurück” kann wieder auf den Startbildschirm zurückgekehrt werden.

1x 5, 3x 8

B. SOLL-Anforderungen

1) Registrieren

Ich als nicht registrierter Nutzer möchte eine Möglichkeit besitzen, mich zu registrieren, um einen Raum eröffnen zu können.

Akzeptanzkriterien:

- Button “Registrieren” leitet an das Formular mit den Angaben weiter
- Formular fragt folgende Werte ab: E-Mail und Nutzernamen (müssen eindeutig sein) und Passwort
- Nach Prüfung, ob die E-Mail oder der Nutzernamen schon vorhanden ist, wird eine Bestätigungsmail versendet
- Erst nach Bestätigung wird der Account freigeschaltet

5, 2x 8, 13

2) Anmelden

Ich als registrierter Nutzer möchte die Möglichkeit besitzen, mich anzumelden, um auf meinen Account zuzugreifen.

Akzeptanzkriterien:

- Button “Login” leitet an das Formular für die Eingabe der E-Mail/Nutzername und des Passworts weiter

- Nach erfolgreicher Eingabe wieder Weiterleitung auf Startbildschirm, jedoch ist jetzt erkennbar, dass man eingeloggt ist

2x 5, 2, 3

3) Abmelden

Ich als angemeldeter Nutzer möchte die Möglichkeit besitzen, mich wieder vom System abzumelden.

Akzeptanzkriterien:

- Button “Abmelden”
- Nach Abmelden Weiterleitung auf Startbildschirm
- Ansicht eines nicht angemeldeten/registrierten Nutzers

C. OPTIONAL-Anforderungen

1) Historie aufrufen

Ich als angemeldeter Nutzer möchte die Möglichkeit, auf eine Historie meiner gespielten Spiele zuzugreifen, um einen Überblick meiner Spiele zu erhalten.

Akzeptanzkriterien:

- Button “Historie”
- Weiterleitung auf Seite mit Übersicht über alle gespielten Spiele
- Zu jedem Spiel steht das Datum sowie die Startzeit und Dauer
- Jedes Spiel ist je nach Ausgang (Sieg oder Niederlage) gekennzeichnet

2x 8, 20, 13

2) Spiel ansehen

Ich als angemeldeter Nutzer in der Historieansicht möchte die Möglichkeit, für ein Spiel eine Wiederholung anzusehen.

Akzeptanzkriterien:

- Button “Ansehen” neben jedem Spiel
- Weiterleitung zu Spiel-Raum
- Keine Interaktion im Raum möglich
- Züge werden automatisch abgespielt
- Button “Zurück”, um das Replay abzubrechen, Weiterleitung auf Historieansicht

2x 13, 5, 8

3) Passwort vergessen

Ich als registrierter Nutzer möchte die Möglichkeit, auf der Anmeldeseite mein Passwort zurückzusetzen für den Fall, dass ich es vergessen habe.

Akzeptanzkriterien:

- Button “Passwort vergessen”
- Weiterleiten auf neue Seite, dort Eingabe von Nutzernamen oder E-Mail
- “Button zurücksetzen”
- E-Mail, in der ein Link ist, der auf eine Seite führt, auf der ein neues Passwort festgelegt werden kann.

2, 3, 2x 5

IV. TECHNISCHES GROBKONZEPT

A. Datenbank - AWS DynamoDB [3]

Eine AWS DynamoDB Datenbank soll zentraler Speicherort sein für alle Daten, wie z.B. laufende und abgeschlossene Spiele, die durch die Anwendung generiert wurden.

B. REST API - AWS API Gateway [4]

Die in der AWS Cloud gehosteten Daten sollen über eine REST API im nötigen Format abrufbar sein. Diese REST API soll dabei durch einen AWS API Gateway realisiert werden, welche je nach Endpoint AWS Lambda Funktionen [5] aufruft.

C. Back-End - AWS Lambda [5]

Alle nötigen Backend-Dienste sollen als AWS Lambda Funktionen realisiert werden. Durch diese werden Daten, wenn nötig transformiert, aus der DynamoDB gelesen und geschrieben.

D. Benutzerverwaltung - AWS Cognito [6]

AWS Cognito soll für eine Benutzerverwaltung genutzt werden, um nötige API Gateway Aufrufe zu authentifizieren und gleichzeitig bereits integrierte Funktionalität, wie Email Verifizierung, bereitzustellen.

E. Web-App – React App [7]

Das Frontend soll React als Framework nutzen, um dem Benutzer eine Website bereitzustellen, welche die gesamte Funktionalität der Anwendung bietet.

F. Deployment - AWS CDK [8]

Um alle nötigen AWS Dienste zu konfigurieren und bereitzustellen soll das AWS Cloud Development Kit für Infrastructure as Code verwendet werden.

[1] Quelle: Wikipedia

[2] <https://app.pro-darts.net>

[3] <https://aws.amazon.com/de/dynamodb/>

[4] <https://aws.amazon.com/de/api-gateway/>

[5] <https://aws.amazon.com/de/lambda/>

[6] <https://aws.amazon.com/de/cognito/>

[7] <https://reactjs.org/>

[8] <https://aws.amazon.com/de/cdk/>