



ZeroSum

Milestone 1

Describe your application and explain how you intend to exploit the characteristics of mobile cloud computing to achieve your application's objectives, i.e. why does it make most sense to implement your application as a mobile cloud application?

The application we will be designing is called **ZeroSum**.

ZeroSum provides a gamified experience for users, whereby they will be able to post questions, allowing the public to answer. However, it's not as simple as just answering the right answer or the answer that is based on your point of view alone. Users that posted the questions are allowed to set the game mode to *Majority* or *Minority*, whereby this game mode will determine the winners of the questions. Winners will be able to gain currency for joining games, experience points and level up.

To tie in together more with gamification, we introduce the concept of collection of hats in this application, whereby there will be a simple storyline introduced to them and they are required to collect all the hats.

These are the hats that the users are able to collect:





The main features of the apps are as such:

Users are able to create their own poll:

- Set topic
- Set voting option
 - No stakes: Players can play for free, without any currency
 - Fixed stakes: Players must put the same amount of currency as other players
 - Limit: Players can only put up to a certain limited amount of currency
 - No Limit: Players can put up to any amount of currency they want
- Set game mode:
 - Majority wins (chosen by default)
 - Minority wins
- Set time limit of the question for people to answer (max up to 24 hr)
- Set public/private matches

They are also able to view all the matches that are going on, viewing different characters; profile.

The intention of such application is more of a social experiment, yet trying to gamify the experience at the same time to provide more context and make things more interesting for the users.

The reason why we intended this to be pushed as a mobile cloud application is because we wanted users to be able to access this application on the go. We wanted it to be an application that is as convenient as possible for users. Users are able to play on the go as well and they can check the status of the questions they have either posted or answer any time they are free. We believe that we should provide

more convenience to the users, especially if we are trying to gamify this social experiment, it should be something that is accessible easily from anywhere.

Milestone 2

Describe your target users. Explain how you plan to promote your application to attract your target users.

Target users

The users we are targeting are people who mainly enjoys messing around with people's minds, in such a that you are trying to confuse people with the question and options that have been posted in the application. People will tend to be confused of what is the "right" answer to choose as what is deemed as the "right" answer may not be the final answer that the crowd has chosen. Hence, users have to be able to determine the answer that they themselves or the crowd may have chosen.

In addition, with the gamification idea, we are incorporating the idea of collection. Hence, this also targets users that likes to collect items (for e.g., games such as Zen Koi, Pokemon etc.). With the incorporation of collection idea, we hope to let users grow to like the mechanic of the application.

Promotion of application

Goals

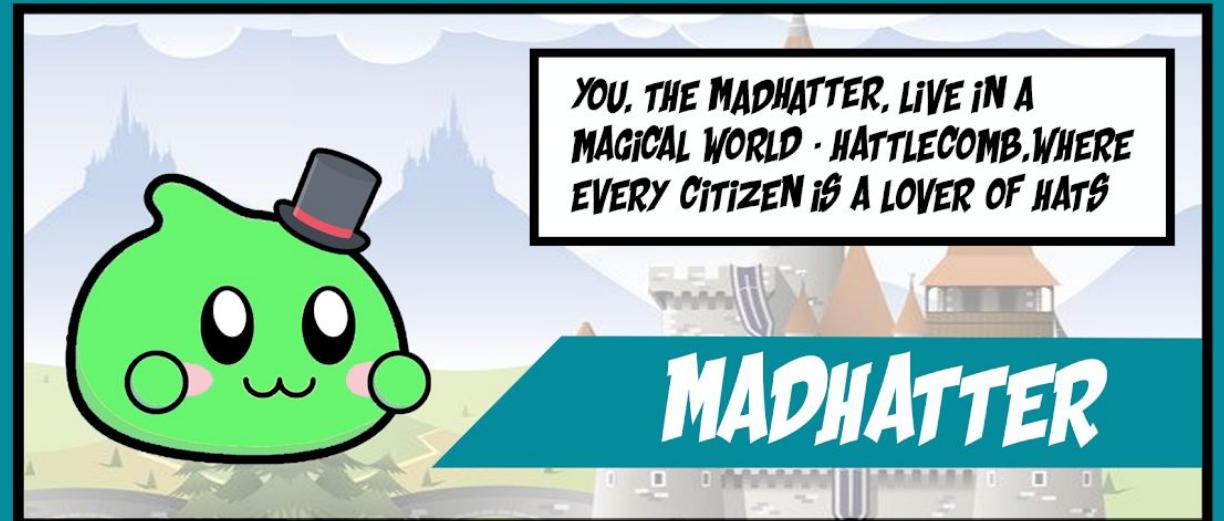
The goal of a successful promotion of ZeroSum, would be to capture the participation from mainly people from the youths who enjoy confusing people and the enjoyment of collection of games. ZeroSum wish to create a platform for these people to ask questions easily, but also a platform for people who enjoys a gamified experience with the collection of hats in mind.

Marketing Strategy

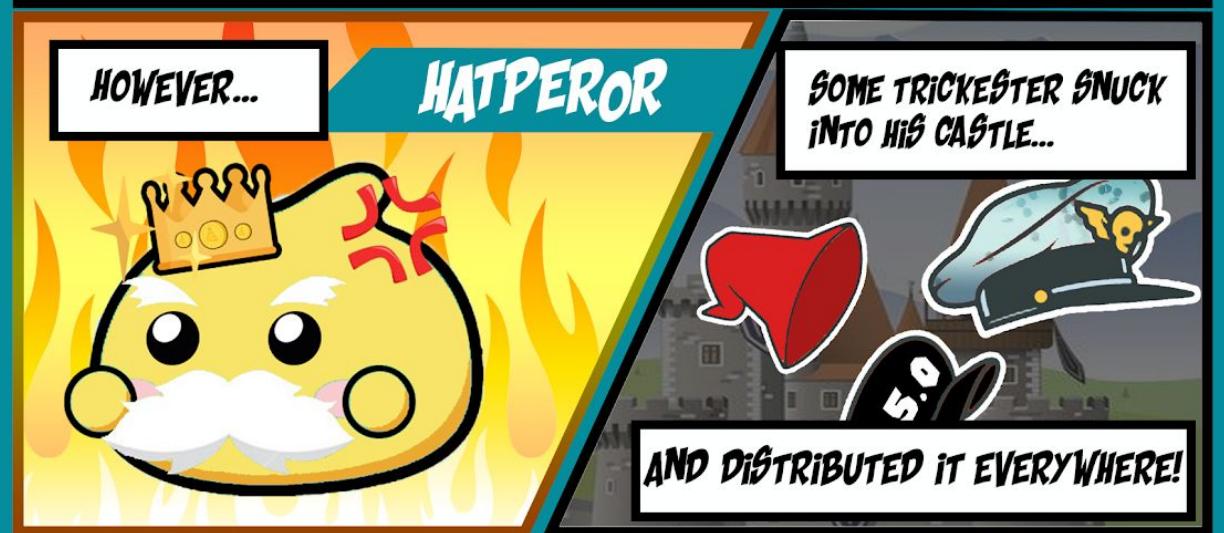
The unique feature about ZeroSum is that it is a gamified platform. Other than the inclusion of the game levelling system and rewards, ZeroSum also introduces characters/mascots in game. Hence, ZeroSum provides a storyline to the users to gain more attraction. The storyline can be made into a **trailer video** or a **comic strip** and the concept is shown below.



1



MADHATTER



SOME TRICKESTER SNUCK
INTO HIS CASTLE...





2

NOW, YOU, BEING THE FAMOUS, MOST
HANDSOME ONE IN HATTECOMB...
YOU SHALL TAKE UP THE MISSION!



HOWEVER, IT'S NOT THAT SIMPLE!! TO
GET BACK THESE MAGICAL HATS...

CS3216

PROVE TO THEM THAT THE HATS
MATTER... PLAY THE GAMES THE HATS
LOVE THE MOST - ZEROSUM!

?

ZeroSum



This trailer video will not be sufficient to explain ZeroSum fully, as it does not explain the main mechanics and features of the application. Hence, there can be a short demo **gameplay** video or comic strip.

In the application, we introduced a character - Hattorial, who acts as the tutorial guide in the application. She can be used as the mascot describe the gameplay in the video/comic strip as well.

The following will be the concept of the video/comic strip:

TUTORIAL



1



HATROOO FELLO HAT!
I AM HATTORIAL, YOUR GUIDE
TO COMPLETE YOUR MISSION!

HATTORIAL

YOUR MISSION IS TO HELP COLLECT
BACK ALL THE HATS!



OUTSMART OTHERS!
CHOOSE THE OPTION
ACCORDING TO THE
GAME MODE!

LEVEL UP! UNLOCK
ACHIEVEMENTS! BUY THE
HATS! ALL THROUGH THE
GAME OF ZEROSUM!



← GAME



Lim Hui Qi



1600

\$ 0 in the pot!

Are dogs cuter than cats?

Majority

Fixed Stakes

5 min

CHOOSE ONE OPTION

Yes

No

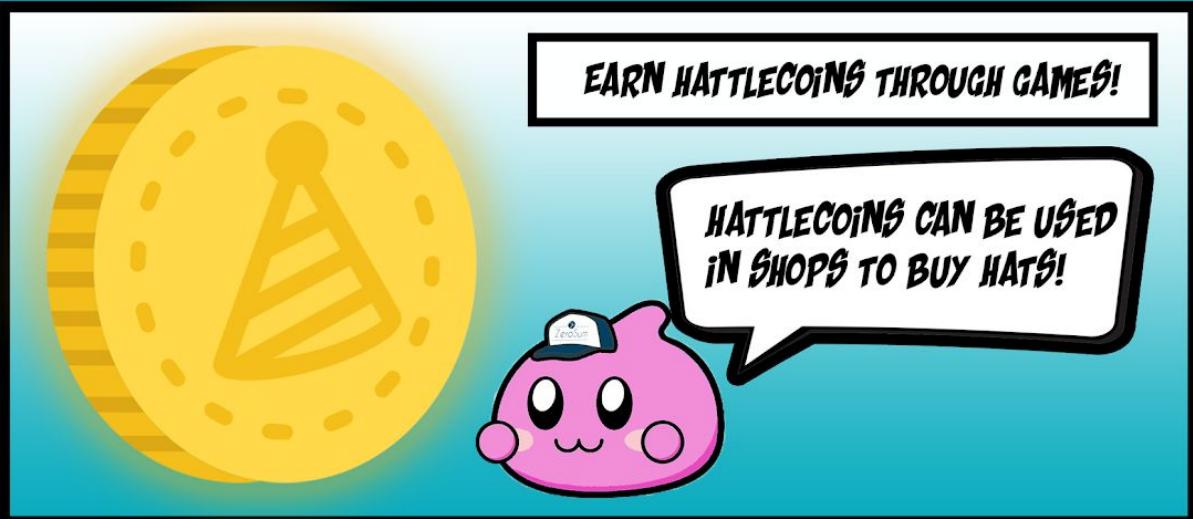
TUTORIAL



2



CHOOSE THE GROUP YOU THINK WILL WIN!



CREATE GAMES WITH A SIMPLE CLICK TO THE BUTTON
RIGHT IN THE MIDDLE OF THE BOTTOM BAR!

TUTORIAL



3

SET YOUR QUESTION
SET YOUR MODE
SET YOUR STAKE
SET YOUR TIME

AND ALL IS DONE. YOUR
GAME IS UP!

GAMES



1 ongoing game!

1600

ALL GAMES

MY GAMES

\$ 100

Are dogs cuter than cats?

Majority

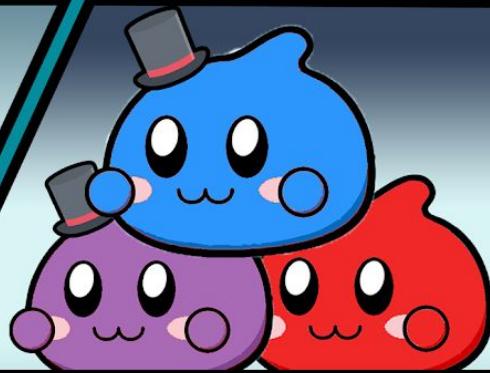
Fixed Stakes

Game ended

JOIN AS MANY GAMES AS YOU WANT. NO LIMIT. EARN 'EM HATS



LOOK AT YOUR OWN PROFILE TO SEE
HOW MANY HATS YOU HAVE



OUTSMART THE OTHERS. LEAD THE
LEADERBOARD. BE THE BEST HAT

WHAT ARE YOU WAITING FOR?

START NOW. PLAY NOW



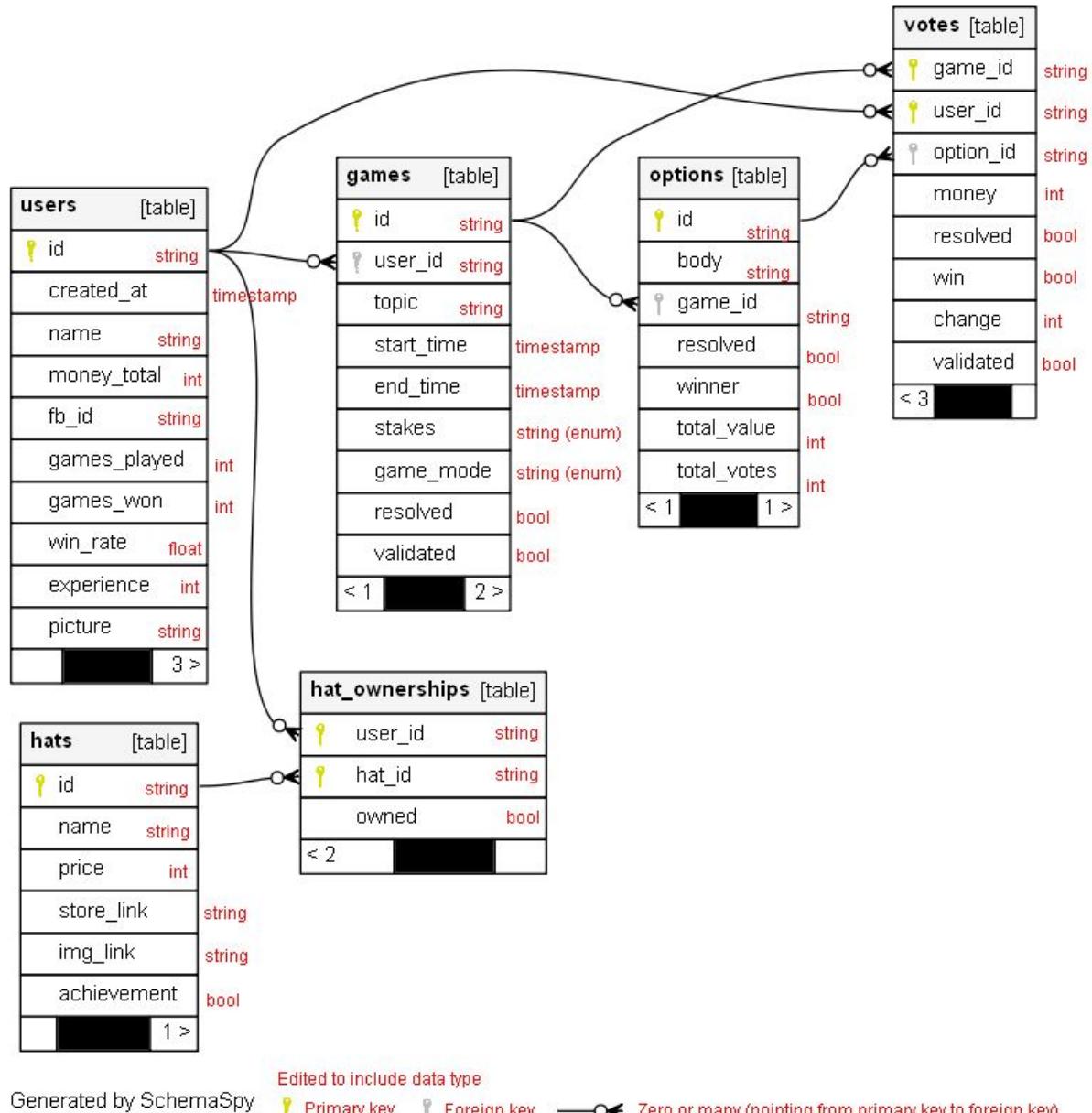
ZeroSum

Visually appealing icons and promotional materials will always tend to attract interest easier, hence the introduction of different character assets in ZeroSum.

Milestone 3

Draw an Entity-Relationship diagram for your database schema.

Database Schema:



Milestone 4

Design and document all your REST API. If you already use Apiary to collaborate within your team, you can simply submit an Apiary link. The documentation should describe the requests in terms of the triplet mentioned above. Do provide us with an explanation on the purpose of each request for reference. Also, explain how your API conforms to the REST principles and why you have chosen to ignore certain practices (if any). You will be penalized if your design violates principles with no good reasons.

In ZeroSum, we used GraphQL instead of REST APIs for our data operations. We decided to use GraphQL is our application is data-driven, with mainly CRUD based client-server communications, and therefore, we feel that GraphQL suits us quite well. The benefits of GraphQL over REST are also well documented - high flexibility, no overfetching/underfetching, strongly-typed schemas.

```
scalar Time

schema {
  query: Query
  mutation: Mutation
}
```

We use Query and Mutation schema for our API. Query is used to retrieve game, shop, leaderboard and profile information from server. Mutation is used to post and update objects.

```

# The query type, represents all of the entry points into our object graph
type Query {
    user(id: ID): User
    game(id: ID!): Game
    activeGames(filter: String!, joined: Boolean, created: Boolean, limit: Int): [Game]!
    completedGames(created: Boolean!): [Game]!
    vote(gameId: ID!): Vote
    votes(limit: Int, after: String): [Vote]!
    gameCount: Int!
    leaderboard(limit: Int!): [User]!
    storeHats(owned: Boolean!): [Hat]!
    achievedHats: [Hat]!
}

}

```

These are our resources that we query on:

- **user**: Used to request for user's information for profile, or to view another user's details
 - Parameters
 - id (*required*): Id of target user. If not provided, defaults to own user
 - Response
 - User
- **game**: Used for request for a specific game's information (not really used in app)
 - Parameters
 - id: Id of target game.
 - Response
 - Game
- **activeGames**: Used to request for a list of games that are active (still ongoing)
 - Parameters
 - filter (*required*): String to search for game by topic title
 - joined: Flag to filter for games that were joined by user
 - created: Flag to filter for games that were created by user
 - limit: Set a limit on how many games to fetch
 - Response
 - List of Game
- **completedGames**: Used to request for a list of games that are completed (resolved), usually used to see results for completed games
 - Parameters
 - created (*required*): Flag to filter for games that were created by user, otherwise only shows joined games.

- Response
 - List of Game
- vote: Used to request for a vote user has placed on a game
 - Parameters
 - gameId (*required*): Id of target game
 - Response
 - Vote
- votes: Used to request for all votes made by user
 - Parameters
 - limit: Set a limit on how many votes to fetch
 - after: Select entries after the vote with id = after
 - Response
 - List of Vote
- gameCount: Used to get the total number of ongoing games
 - Parameters
 - Response
 - Number of ongoing games
- leaderboard: Used to retrieve an ordered list of top players sorted by win rate, only players that have played more than 10 games will be shown
 - Parameters
 - limit (*required*): Set a limit on how many players to fetch
 - Response
 - Ordered list of User
- storeHats: Used to retrieve a list of hats available on store
 - Parameters
 - owned (*required*): Flag to return hats bought by user if true and hats unowned by user if false
 - Response
 - List of Hat
- achievedHats: Used to retrieve a list of hats tied to achievements that user has cleared
 - Parameters
 - Response
 - List of Hat

```

# The mutation type, represents all updates we can make to our data
type Mutation {
  deleteUser: Boolean!
  addGame(game: GameInput!): Game
  addVote(vote: VoteInput!): Vote
  buyHat(id: ID!): Hat
  validateResult(gameId: ID!): Boolean!
}

```

These are our mutations:

- `deleteUser`: Removes user from records
 - Parameters
 - Response
 - Boolean that shows if operation is successful
- `addGame`: Allows user to create a game
 - Parameters
 - game (*required*): <See GameInput>
 - Response
 - Created Game
- `addVote`: Allows user to make a vote on a game
 - Parameters
 - vote (*required*): <See VoteInput>
 - Response
 - Created Vote
- `buyHat`: Allows user to ask to purchase a hat
 - Parameters
 - id (*required*): Id of target hat
 - Response
 - Purchased hat
- `validateResult`: Allows user to acknowledge results of completed games, so that they are removed from their list
 - Parameters
 - gameId: Id of target game
 - Response
 - Boolean that shows if operation is successful

```



```

These are the input types used for the mutations in the previous section.

- **GameInput**
 - topic (required): Topic for the game
 - duration (required): Duration in minutes that the game will be active for
 - gameMode (required): Game Mode for the game <See GameMode>
 - stakes (required): Stakes Mode for the game <See Stakes>
 - options (required): List of text options for the game
- **VoteInput**
 - gameId (required): Game to be voted on
 - optionId (required): Option that user chooses
 - amount (required): Amount of money user places on the vote

```

enum GameMode {
    MAJORITY
    MINORITY
}

enum Stakes {
    NO_STAKES
    FIXED_STAKES
    FIXED_LIMIT
    NO_LIMIT
}

```

These are the enum types used

- **GameMode**
 - MAJORITY: Option with most amount of money wins
 - MINORITY: Option with least amount of money wins

- Stakes

- NO_STAKES (not used): Votes do not require any money
- FIXED_STAKES: Votes require a server-defined amount of money (100)
- FIXED_LIMIT (not used): Votes can only be up to a user-defined limit
- NO_LIMIT: Amount of money placed on a vote is not restricted

The types below define the different return types that the client receives from queries and mutations

```
type Game {  
    id: ID!  
    owner: User  
    topic: String  
    startTime: Time  
    endTime: Time  
    totalMoney: Int  
    gameMode: GameMode  
    stakes: Stakes  
    voted: Boolean  
    resolved: Boolean  
    options: [Option]  
}
```

- Game

- id: Id for the game
- owner: Creator of the game
- topic: Topic question of the game
- startTime: Time at which game was created
- endTime: Time at which game is stipulated to end
- totalMoney: Total money in the pot for the game
- gameMode: Game Mode for the game
- stakes: Stakes Mode for the game
- voted: Whether the user has voted on this game
- resolved: Whether the game has been resolved by server (game ended)
- options: Options tagged to the game

```
type Hat {
    id: ID!
    name: String!
    price: Int!
    img: String!
}

type Option {
    id: ID!
    body: String
    result: OptionResult
}

- Hat
  - id: Id for the hat
  - name: Name of the hat
  - price: Price of the hat (if applicable)
  - img: Image of the hat (mystery image if not owned)
- Option
  - id: Id for the option
  - body: Text of the option
  - result: Result of the option after game is resolved <See OptionResult>
```

```

type User {
    id: ID!
    name: String
    img: String
    money: Int
    winRate: Float
    level: Int
    expProgress: Float
    ranking: Int
}

type Vote {
    game: Game
    option: Option
    money: Int
    resolved: Boolean
    result: VoteResult
}

```

- User
 - id: Id for the user
 - name: Name of the user
 - img: Profile image of the user
 - money: Money owned by user
 - winRate: Win rate for games for user
 - level: Level of the user
 - expProgress: Exp progress to the next level as a fraction
 - ranking: User's ranking on the leaderboard
- Vote
 - game: Game that the vote is attached to
 - option: Option that user voted on
 - money: Amount of money placed on the vote
 - resolved: Whether the vote has been resolved by server (game ended)
 - result: Result of the vote after game is resolved <See VoteResult>

```

type VoteResult {
    win: Boolean
    netChange: Int
}

type OptionResult {
    voteCount: Int
    totalValue: Int
    winner: Boolean
}

- VoteResult
  - win: Whether the vote was part of a winning option
  - netChange: The amount of money gained or lost by user from the vote resolution

- OptionResult
  - voteCount: Number of votes under this option
  - totalValue: Total amount of money placed on this vote
  - winner: Whether the option was a winning option

```

Comparison of how our use of GraphQL matches up with REST principles

Principle	GraphQL
Client-Server	Our GraphQL schema was designed as a contract between what the client side of our application requires and what the server can provide. Both parties are able to develop separately using only the schema as a reference.
Stateless	Queries and mutations made to the GraphQL server are stateless, and can be run purely with the information provided in the requests.
Cacheable	Caching for our GraphQL queries is handled by the Apollo Client, where we are able to cache queries and serve cached information in cases of poor network
Layered System	We have not used a layered architecture for this application, but our implementation does not expose any information about how each request is processed, and should be able to work in a layered system
Uniform Interface	This is less relevant to our use of GraphQL as the schema provides clear and adequate information to the client about

the resources available to them, the parameters required and the responses given. We have provided some additional schemas that appear to give the same type of response (e.g. list of hats). However, the nature of the queries are often different and require different query parameters (querying for hats available for purchase in store is different from querying for hats from completed achievements). As such, we have separated some queries to make it clearer to the client about what they provide.

Milestone 5

Share with us some queries (at least 3) in your application that require database access. Provide the actual SQL queries you use (if you are using an ORM, find out the underlying query and provide both the ORM query and the underlying SQL query). Explain what the query is supposed to be doing.

Query 1: Obtaining the hat_ownership entry that relates the user to a hat, can be used to verify that user owns the hat

GORM command:

```
hatOwnership.HatId = "f8078387-6726-422b-83cc-6d139be7a795"
hatOwnership.UserId = "1ApriYiWdJvVYsgq5BQjP3BdC4D"
db.Where("hat_id = ? AND user_id = ?", hatOwnership.HatId,
hatOwnership.UserId).First(&foundOwnership)
```

Corresponding SQL Query:

```
SELECT * FROM "hat_ownerships" WHERE (hat_id =
'f8078387-6726-422b-83cc-6d139be7a795' AND user_id =
'1ApriYiWdJvVYsgq5BQjP3BdC4D') ORDER BY "hat_ownerships"."hat_id" ASC LIMIT 1
```

Query 2: Getting list of top 10 players sorted by win rate, where only users who have played at least 10 games are considered

GORM command:

```
minGames = 9
limit = 10
db.Where("games_played > ?", minGames).Order("win_rate
desc").Limit(limit).Find(&users)
```

Corresponding SQL Query:

```
SELECT * FROM "users" WHERE (games_played > '9') ORDER BY win_rate desc  
LIMIT 10
```

Query 2: Getting the total number of active games at the current time

GORM command:

```
time.Now() is 2018-09-28 14:38:17  
db.Model(&models.Game{}).Where("end_time > ?", time.Now()).Count(&total)
```

Corresponding SQL Query:

```
SELECT count(*) FROM "games" WHERE (end_time > '2018-09-28 14:38:17')
```

Milestone 6

Create an attractive icon and splash screen for your application. Try adding your application to the home screen to make sure that they are working properly. Include an image of the icon and a screenshot of the splash screen in your writeup. If you did not implement a splash screen, justify your decision with a short paragraph. Add your application to the home screen to make sure that they are working properly. Make sure at least Safari on iOS and Chrome on Android are supported.

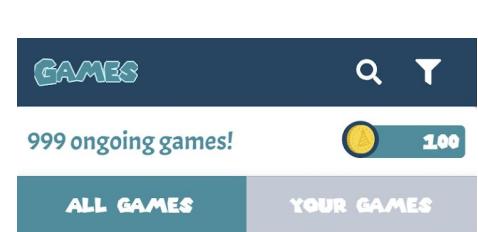


Milestone 7

Style different UI components within the application using CSS in a structured way (i.e. marks will be deducted if you submit messy code). Explain why your UI design is the best possible UI for your application. Choose one of the CSS methodologies (or others if you know of them) and implement it in your application. Justify your choice of methodology.

General Design

For the idea of the application, we gamified the user experience and the idea, and our design will appear more gamified as compared to the usual web applications. Hence, there will be several icons and fonts that are meant to fit more into the game theme in order to let users immerse themselves better with the application. In general, the design of the application will always have the **top** and **bottom** bar.

	<p>Top Bar</p> <p>The purpose of the top bar is to always let the users know what page are they on, since there will always be label at the top left corner of the page. Depending on the page, there</p>
---	---

may or may not be buttons at the top right corner of the page. This will be further elaborated later when we talk more about the specific screens. The title of the screen is a font that will let the users feel that they are in a game. The name is being outlined with a bright color, to give the title a more eye-catching look. This is to allow users to catch on quickly what page are they currently on right now.

The positioning of the name is positioned such that users will be able to see the name immediately upon loading any page, as it will be right at the top of the page and an obvious color to them.

Bottom Bar

The bottom bar acts as a form of navigation for the users. For the page the users are currently at, users will be able to see the icon being “pressed down”, indicated by the darker shade of blue behind the icon.

The icons chosen are also meant to be intuitive and straightforward for the users to understand and navigate around without any guidance.

From the left to right, the icons represent *Home, Shop, Add, Leaderboard, Profile*. The icons are also chosen to look consistent with each other to look more coherent.

Home Page

Upon entering the home page, users will be able to see the page, and some cards on the screen, whereby each of the card represents a separate game on its own. As mentioned in the first segment of *General Design*, titles are all aligned to the top left, with a font that will give users a stronger feeling of being in a game. In this page, there are 2 additional buttons on the bar - the search and filter buttons. Icons are placed at the corners since the essential buttons are placed at the top and corners for better user experience.

Right below the top bar, users are able to see the number of ongoing games and the amount of Hattlecoins (currency) they currently have. Position is as such is to give a task-focused approach and users know the important information are at the top. Hence, the users are also able to view their current tabs (Your Games and All Games) at the top due to the same reason, and the **layout of the tabs is consistent throughout the whole application.**

	<p>Individual Cards</p> <p>All the cards and base buttons are designed to be the same color in the whole application, which is a white base. This gives users an indication of what are considered buttons, thus they know what to press.</p> <p>There are icons to represent the different game modes in order to improve readability, allowing users to absorb information much quicker as compared to all words. The fonts chosen for the main title and the description are also meant to complement each other, but the description font <i>gives a less dominating effect</i>, so users will know which is the main focus.</p>
	<p>Creating Game</p> <p>The page is split up into different sections by using the header on this page, giving a clear distinction to users of each section. This creation of games is supposed to give a similar feeling of Google Forms, but with more gamification feeling instead of simple radio buttons.</p> <p>Topic</p> <p>The input of the topic is meant to be straightforward, just like usual UI so it provides a simple way for the users to input in their question.</p> <p>Options</p> <p>Options displays two different things, <i>Type here</i> and <i>NEW OPTION</i>, to indicate that users are able to tap on them and input in the desired input. When users press on <i>NEW OPTION</i>, a new, darker shade of blue option pops out, showing the differences in older and newer options. This is to allow users to differentiate and input easily, but also having a more dynamic interface to avoid confusion. [If 4 boxes were displayed at the start, users may think it's compulsory to key in 4 options. Thus, the dynamic addition of boxes and maximum up to 4 options allows them to understand easily.]</p>

	<p>Game Mode, Stakes, Time</p> <p>For the rest of the modes, they are displayed and a rectangular box will be shown at the back as a form of selection of that option. This is to allow them to see more options, and a more visual display as compared to dropdown or checkboxes. This is also to tie in more together with the concept of a game interface as well.</p>
	<p>Games Room</p> <p>Upon entering a game room, be it from the one you or others have created, users will see this screen.</p> <p>They will be able to see how much Hattlecoins they have, which has a same look as the main screen. The icons used on this page are the same as the main home screen/games page, to ensure consistency so the users will understand what do the different icons represent.</p> <p>The base color for the options are of a different color (same shade of blue in the home screen), as to allow consistency for the theme and design of the entire application, but also to allow it to act as a divider between the topic/question from the options, so the users are able to distinguish them easily.</p> <p>Similar to the cards in the home screen, the options are presented with a white base as well. Similarly, this is to ensure users know that they are clickable and they can pick one option.</p>

You're about to bet 100 HattleCoins.

HattleCoins are not refundable after submission!

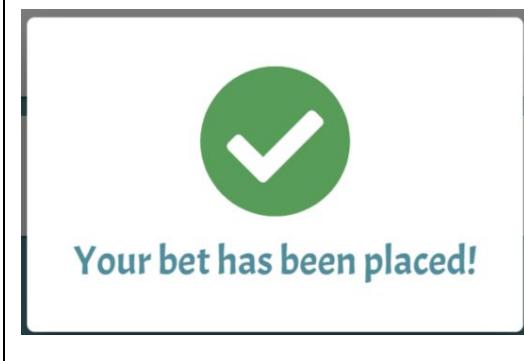
SUBMIT

How many HattleCoins to bet?

HattleCoins are not refundable after submission!



SUBMIT



Confirmation pop-ups

After choosing the desired option, pop-ups will appear as a form of confirmation. To allow the users to understand what form of currency they are using to bet in the game, the same icon for Hattlecoins is being used here as well.

If the betting is successful, a green tick will appear as a form of confirmation, allowing them to understand that it has been processed.

The reason of using pop-ups but not navigating to a new screen is to reduce the amount of disturbance to the users, and to allow users to return back to select other option easily on the same screen. Hence, this reduces transition time between screen, allowing user to make the *redo* action faster, as it's also important for users to *redo* and *rectify* any actions/mistakes they make.

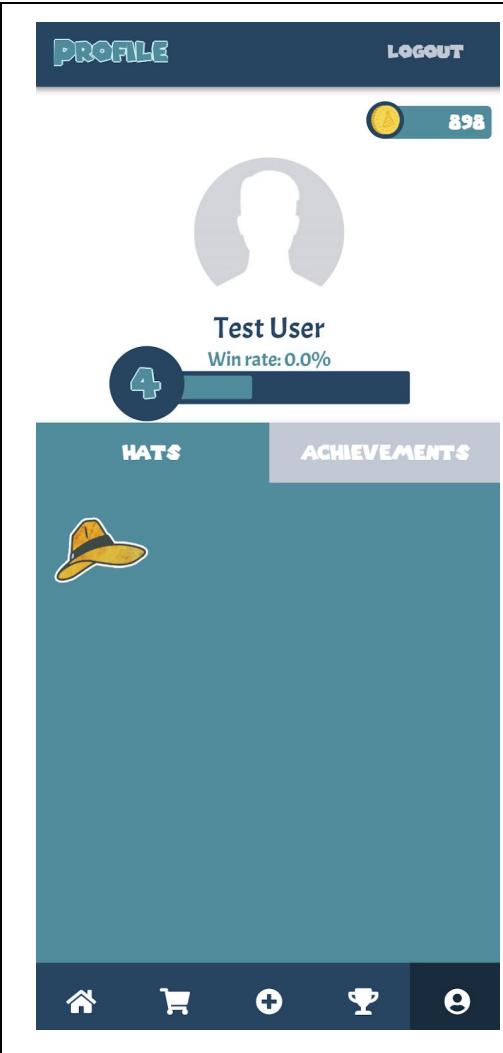
Confirmation screen

Upon confirming the option, the user will see a relative similar screen as compared to before, with a slight difference - the option they have chosen is now larger, while the other options will now be smaller and greyed out. This is to indicate and emphasize the option they have chosen and they can no longer select other options. The Hattlecoins displayed on the top right corner also shows the amount they have betted. The title also have changed to "Vote Submitted".



Results page

The results page still looks similar to the game room page in order to let the UI remain consistent. The main difference now is just the UI will show the winning result by letting it have a white base, and the rest of the options **remain greyed out**, with the percentage at the side for the users to refer to. It will also reflect the amount of coins the user have won if they have chosen the right choice. The color chosen for all these options is to *allow a color differentiation* for the users, and letting them understand which one is the focus and winning option.



Profile page

The profile page is meant to be kept simple for the users. Only certain key information are displayed to the users. The *static information* (those that are essential to the users as compared to other information) remains at the top half of the page. Instead of just displaying numbers, information such as the experience points for the levelling system is shown as a bar, to give a more visual representation, sense of satisfaction and a more refined UI. For the rest of the information which are the collection of hats and achievements, they are being categorised into 2 different tabs.

The users are able to see them the hats and achievements being aligned in an invisible grid style to have a neater structure overall. There is also a **white outline** around the hats and achievements to give a more outstanding effect.



Shop

The shop's UI is relatively straightforward. It has a similar UI from the other screens, whereby users are able to see the amount of coins they have at the top right corner. Similarly to betting in games, the same icon for the coin is being used in the shop, allowing users to understand that the same type of currency can be used here.

The base card colors for the shop is not white, unlike the other pages. The reason behind is to *differentiate* the game questions and options with the in-game items (hats), creating a distinguishing factor between both options.

In addition, the choosing of colors was also based off the whole theme and color coordination. The hats displayed should be consistent with the look in the profile page, which is a white border. Hence, if a white base is being used, the **white border** will not be shown and the look of hats will not be **consistent** if a different border color was used.

The background color of the page was also meant to be kept consistent with all other pages - the same shade of blue. This is to **ensure consistency** of the theme and design, and not to have any jarring designs out of nowhere to the users.

LEADERBOARD		
GLOBAL	FRIENDS	
RANK	NAME	WIN RATE
1	Owonihaoitme!	89.3%
2	Owonihaoitme!	89.3%
3	Owonihaoitme!	89.3%
4	Owonihaoitme!	89.3%
5	Owonihaoitme!	89.3%
6	Owonihaoitme!	89.3%
7	Owonihaoitme!	89.3%
8	Owonihaoitme!	89.3%
9	Owonihaoitme!	89.3%
99	Owonihaoitme!	89.3%

Leaderboard

There are 2 different categories for leaderboard - Global and Friends (Since this application requires you to be linked to Facebook). The rest of the information such as the amount of coins you have is not crucial for this page, and hence it is not being displayed. Similarly to the home and profile page, since there are 2 different tabs, the same UI is being used to ensure consistency.

Right at the top of the page, under the tabs, there will be a header shown that states *rank*, *name* and *win rate*. This is done instead of putting the numbers in the card itself as to reduce the amount of information squeezed into the card. In addition, the entire page will have **consistent** information (meaning it will still be rank, user name and win rate no matter what values are they), hence it will be better to **extract out** the common information and generalize it for the entire page. Thus, the creation of **header** at the top. To let users understand the ranking page better, those that are currently on the leaderboard will be shown as *rounded rectangles* on the page, and a *different color for the rank number and win rate* will be used. For their **personal rank**, it will be shown at the bottom of the page with a **non-rounded rectangle**, whereby the information displayed is also of a **darker shade of blue**.

To make the leaderboard page better, the ranking for the top 3 are of different color and design. This is to allow users to differentiate and detect the top 3 ranks easily. There are no other special differentiations for other ranks below 3 as there is no need to display such information and create changes, as the **essential** information in this page are mainly the *top 3*.

Justification of CSS Methodology

We chose to adopt CSS in JS firstly because it is compatible with the libraries we are using such as React and Material-UI. But besides that, it also offers scoped styles where styles are bound to the scoping rules of surrounding code. This allows us to

not worry about CSS selectors clashing in the global scope. And it only takes an import to make a rule available to the rest of the application. Another huge benefit that it provides is in-built rendering of critical CSS. This replaces the need to manually maintain and optimize critical CSS as CSS-in-JS enforces this rule to improve initial load times.

Milestone 8

Set up HTTPS for your application, and also redirect users to the https:// version if the user tries to access your site via http://. HTTPS doesn't automatically make your end-to-end communication secure. List 3 best practices for adopting HTTPS for your application. Explain the term "certificate pinning" and discuss the pros and cons of adopting it, as well as justify your choice whether or not to use it in your app.

3 best practices

- HTTPS protects the integrity of the website. It prevents intruders from tampering with the communications such as injecting advertisements into websites which can potentially break our user experiences and create security vulnerabilities.
- HTTPS enable offline app experiences with the building of progressive web apps, require explicit permission from the user before executing. HTTPS is thus a key component to the permission workflow for the new features. In addition, HTTPS can optimize performance, thus making the webpage to load faster.
- HTTPS can also protect the online privacy of the users. In ZeroSum, users are required to login into the page via Facebook, and it is necessary to protect their privacy, since Facebook contains a lot more of their personal and sensitive information as compared to ZeroSum.

Certificate Pinning

The client can authenticate the server that it is connecting to by validating that it has a server certificate issued by a Certificate Authority (CA) that the client trusts. Certificate Pinning provides a framework for reducing the tendency to rely on trusting third parties when making security decisions concerning identity. It creates an independent "whitelist" defining what a trustworthy digital certificate or encryption key should look like.

In this way, the server can be identified from its public key, without looking to a third party for verification. This provides a layer of protection against certificate forgery, since the trusted certificates are likely to be controlled by yourself instead of a third party.

However, this also means that the aforementioned public key, or cryptographic identity, has to be stored on the server, and if these are for some reason lost, then no client will be able to authenticate the server and access the website.

Justification of not using in the application

We used the third party certificate authority letsencrypt for our SSL certificates, which has also become the de-facto standard for modern apps and websites. Due to their strong reputation and track record, the security risk of them being compromised is minimal, and not worth the effort of implementing certificate pinning on our own.

Milestone 9

Implement and briefly describe the offline functionality of your application. Explain why the offline functionality of your application fits users' expectations. Implement and explain how you will keep your client synchronised with the server if your application is being used offline. Elaborate on the cases you have taken into consideration and how they will be handled.

Players are able to check their profile still. It is important for users to be able to minimally check their profile as it should be a status that they are able to know at any point of time.

All these information will be cached so the players will be able to access it offline as well.

Milestone 10

Compare the advantages and disadvantages of token-based authentication against session-based authentication. Justify why your choice of authentication scheme is the best for your application.

	Token-based Authentication	Session-based Authentication
Performance	The backend need only verify the token signature to authenticate the user. User id and permissions can be stored on the token	A look-up for the session id has to be performed on the backend database for every user request. Another lookup then has to be made for the user data (e.g. permissions)
State of data	Stateless. The tokens are only stored client-side. The	Stateful. Session data on (with each user possibly having

	backend server doesn't store any information about the user session at all.	multiple) is stored on the backend. This makes it less scalable, and session management across multiple domains can be quite a challenge.
Mobile Ready	Easier to implement on both iOS and Android. Tokens are also easier to implement for Internet of Things applications and services that do not have a concept of a cookie store.	Native mobile platforms don't really mix well with session-based authentication.
Scalability	Tokens are stored client-side, so there will be no scalability issues for the backend server.	Sessions are stored in the server's memory, scaling becomes an issue when there is a huge number of users using the system at once. Deployment complexity will increase due to the need for load-balancing and session persistence systems to ensure that session data are not lost due to isolated server node crashes.
Size	Tokens are very large compared to session ids. They have to be sent on every server request, so there is a limit to how much data can be stored on a token	Session ids are small (compared to tokens). The heavy lifting is done server side

Justification:

We chose token-based authentication for our application for simplicity. Tokens are widely used in modern app development and are relatively easier to implement. We also do not have functionality that require tracking individual sessions. Finally, as a PWA targeted towards mobile platforms, token authentication allows us to avoid dealing with tricky issues surrounding cookies on mobile devices.

Milestone 11

Justify your choice of framework/library by comparing it against others. Explain why the one you have chosen best fulfils your needs. Lastly, list down some (at least 5) of the mobile site design principles and which pages/screens demonstrate them.

Framework chosen and reason for choosing

We explored 3 particular libraries before coming to a decision - Material-UI, Ant-Design, Onsen UI. Although Ant Design provides very elegant UI components, we felt that it did not provide the kind of mobile experience that we were looking for. As for Onsen UI and Material-UI, we tried out both libraries before coming to a decision. While Onsen provides most of the mobile UI components that we need, we felt that Material UI had better documentation and support as it has a larger ecosystem of users and developers. Therefore, we chose Material-UI.

Mobile Site Design Principles

The screenshot shows the home screen of the ZeroSum app. At the top, there's a dark header bar with the word "GAMES" on the left and a search icon on the right. Below the header, a teal banner displays "999 ongoing games!" and a gold coin icon with the number "100". There are two tabs: "ALL GAMES" (selected) and "YOUR GAMES". The main content area contains four identical-looking game cards, each with a yellow dollar sign icon and the number "99999". The text on each card reads: "Has everything been here forever, or when did it begin to exist?". Below the text, there are three icons: a blue square with a white diamond (Majority), a green square with a white circle (Fixed Stakes), and a red square with a white triangle (23h 39min). A footer navigation bar at the bottom includes icons for Home, Cart, Plus, Trophy, and User.

Optimized For Mobile

In ZeroSum, the interface is **optimized for mobile** as much as possible.

- There are no horizontal scrolling necessary to see any primary content. In most of the pages, especially the home page, they are all vertical scrollable.
- The content are optimized to fill up most of the screen, removing many elements that could distract users from the content. The screens are mainly all kept simple to avoid cluttering of information.
- The layout of the columns are also reduced, mainly to just single column site layout, as seen in the home page.

How much Hattlecoins to bet?

Hattlecoins are not refundable after submission!



99999

SUBMIT

Making Calls-To-Action Buttons Mobile Friendly

Users tend to miss out UI elements easily, so the key calls-to-action are obvious enough for the users to see them.

For instance, in the page that users are supposed to input the amount of coins, they are able to see this pop up appearing, with a input box, submit button and a cross button. All other UI elements on the screen are blacked out and not clickable, as to allow the important buttons to be easily clickable, and other elements on the page is not interfering with the buttons.

How much Hattlecoins to bet?

Hattlecoins are not refundable after submission!



99999

SUBMIT



Design Finger-Friendly Touch Targets

The size of the buttons are sized relatively large and appropriate for the users to click on. In addition, the relative distance (margin) between the touch elements are at least 10px apart, to reduce the error of users accidentally tapping on the wrong button with their finger. For instance, the *submit* and the *cross* buttons are designed to be large and center of the screen, so users will not be able to miss them and tap them without having high margin of errors. The bottom bar of the application have 5 different icons - but we ensure that they are all at least 10px apart, so the users are able to tap the right button and reduce the margin of error.

In addition, our options are purposely designed to be large of a surface area, so users are able to have more flexibility in terms of the tapping area.



Using High Quality Assets

Since our platform have different assets to fit into the game theme more, there is even a greater need to provide high quality assets for greater user experience.

For instance, our assets such as the currency (Hattlecoins) and hats are designed to be high quality, as they are the products of our application itself. This will then be able to capture the attention of site users who may skim, and drive call-to-actions.

GAMES

Owonihaitme!

999999 in the pot!

Blablablablablablablabla
blablablablablablablabla
blablablablablablablabla
blablablablablablablabl?

Majority Fixed stakes 23h 39min

CHOOSE ONE OPTION

Blablablablablablabla
blablablablablablabla

Blablablablablablabla
blablablablablablabla

Blablablablablablabla
blablablablablablabla

Blablablablablablabla
blablablablablablabla

Users do not need to Pinch-To-Zoom

No matter which screen the users is on, there is no need for them to pinch to zoom in. Users are usually frustrated when they need to zoom in and out in order to read a text or see an image and they may miss out important details. Hence, all the screens are designed such that the users won't ever need to change the size.

ZeroSum ensures the text and images to be properly sized, as to allow users to view the screens in their natural state.

CREATE GAME

Topic

I want to ask...

Options

+ NEW OPTION

Game Mode

Majority Minority

Stakes

No Stakes Fixed Stakes
Limit No Limit

Time

5m 30m 1h 3h
6h 12h 18h 24h

SUBMIT

Choose the simplest input method for each task

Users will normally find it easier to tap toggle icons than to enter text or select from a dropdown. The key design considerations here will be making the interface here easy for the users to tap and find their desired options.

The options are designed to be dynamic increasing/decreasing instead of clicking any dropdown box to select how many options (ranging from 1 - 4). This is to reduce the number of clicks for the users, improving their user experience.

For the rest of the modes, the options are displayed and arranged in such a way that the space used is optimized, and users are able to view all the possible options at the same time. This design will be better as compared to choosing options from a dropdown menu, as users will not be able to see everything right at the moment they enter the screen. This also reduces the number of taps, improving the *redo* and *undo* actions the users may have.



Make it easy to get back to the home page

When users navigate through the site, they would want an easy way to get back to the initial homepage. Most of the screens will always have this bottom bar, whereby the home page is indicated by the house icon at the bottom left corner. Hence, users are able to navigate back to the home page easily with just a click on the icon. The home icon is being used as it is regarded as a more intuitive icon to "home" since it is a house.

Milestone 12

Describe 3 common workflows within your application. Explain why those workflows were chosen over alternatives with regards to improving the user's overall experience with your application.

Gameplay

Creating questions

To create a game/question, users are required to click the **Add** button on the screen, whereby they would have to fill up the form shown and **submit** at the end. This is made convenient for the users since the add button is at the persistent bottom bar of the page, whereby the add button is located in the middle. This makes it accessible to the users to create questions at any point of time from other locations, instead of giving extra steps to just create a question.

The initial design for adding options was a dropdown menu. It included fields for Topic (text input), Options (dropdown with 'NEW OPTION'), Game Mode (radio buttons for Majority and Minority), Stakes (radio buttons for No Stakes and Fixed Stakes), and Time (radio buttons for 5m, 30m, 1h, 3h, 6h, 12h, 18h, 24h). A 'SUBMIT' button was at the bottom, along with a bottom navigation bar with icons for Home, Cart, Plus, Trophy, and User.

Looking at the current design, the initial design for the adding of options was chosen to be a dropdown list to select how many options were there. According to the number of options, the corresponding number of rectangles will appear. The image below shows the initial planning of our design.

A planning sketch of the 'CREATE GAME' form. It features a 'Topic' field with placeholder text 'To the people out there, what is the meaning of life?'. Below it is a 'Options' section with a dropdown menu showing the number '4' and four items labeled 'FOOD'. The 'Game Mode', 'Stakes', and 'Time' sections are identical to the initial design. The 'SUBMIT' button is at the bottom, along with the same bottom navigation bar.

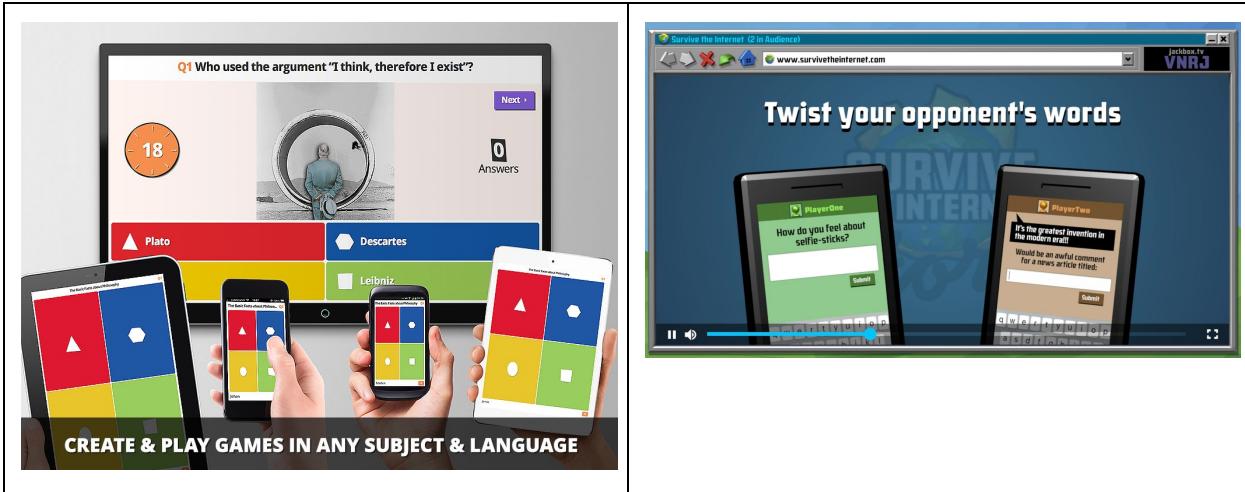
However, we found that the whole design looks very incoherent, and the

design of the dropdown box makes the whole process slower and there were too much disturbance to the users upon opening the dropdown box. Hence, the final design was a design that was more dynamic, so that users can remove and add options when they wise to do so instead of having an extra step to change the number of options. Thus, this new design will thus be a more convenient, aesthetically wise nicer and more flexible for the users, thus improving the usability of the application and user workflow.

Answering questions

Answering of questions can be considered as one of the key workflows in ZeroSum. Users are required to **choose an option, inputting the currency to bet**, and lastly confirming their desired choice. For ZeroSum, the answering of questions is made to have a simplistic look, whereby the topic/question is at the centre and top of the page, and the options are layout over the other half of the page. This is to make the selection of options easier and reduce errors made by the users due to the size and design of the buttons. After choosing an option, instead of navigating the users to a new page or a window, a pop up appears instead as to reduce the transition time and disturbance caused to the users. After confirming the status prompted by the pop up, users can simply close the window by tapping anywhere else on the screen. For the answering of questions, we were taking reference with respect to more of existing games such as Jackbox party games and Kahoot.

Some screenshots of Jackbox party games and Kahoot interface can be shown below:



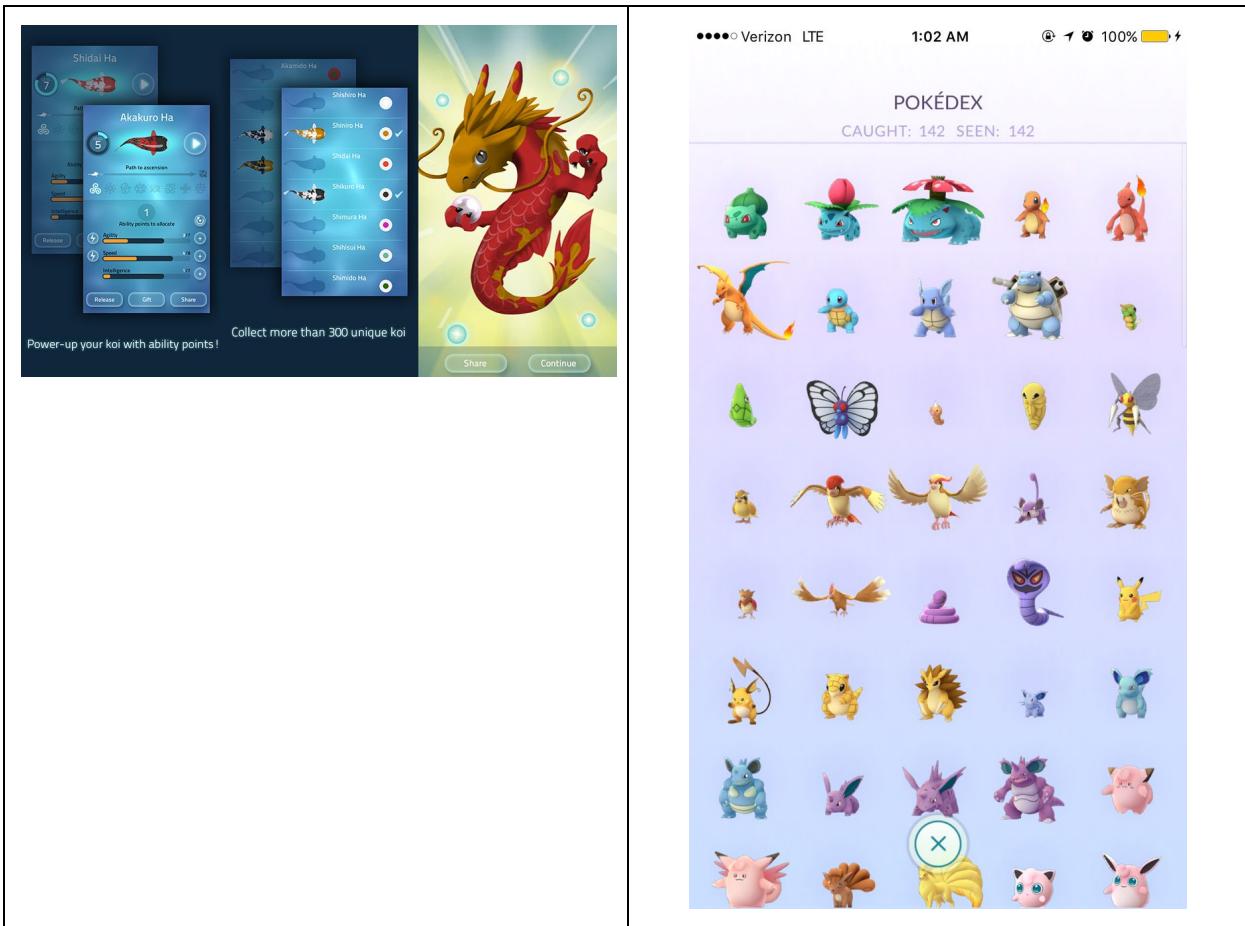
Hence, we were inspired by such designs as reference, and tried to optimize and make use of the space available to show the options as much as possible, and having a nice and presentable UI at the same time.

Hence, to have a presentable and good workflow for the users, we would want to ensure the usability of the application, allowing them to rectify and view information with as minimal error as possible. Thus, basing off popular games and platforms, we came up with our final workflow design to improve the overall experience. We also tried to make sure the look in the application are consistent, so to make the entire application more coherent and pleasant, as we believe that this will improve the overall experience to a greater extent.

Checking Profile and Hats

Users are able to check their profile to see their level, experience, win rate and collection of hats. This is treated as a common workflow for users as the profile page is the only place the users are able to check their statistics and collection of hats. As this platform is gamified, the users would want to level up as fast as possible to "clear" the mission, hence they would want to see their level and experience points. In addition, people that like to collect hats will see their collection of hats quite often as to get the sense of satisfaction and achievement.

For the layout of the hats, the idea of having a grid system viewing came out right at the start. Looking at some games such as Zen Koi collection and the Pokedex in Pokemon Go...



We do realize that in terms of layout wise, the first thing to take into account is making the icons large enough for users to view. In addition, we also feel that by having a grid based system to view the collection of hats, they will be arranged in a much neater way, and more hats can be displayed within that page. Hence, it's based off largely on existing platforms for us to decide on the grid display system of the collections, especially it has been working for games like Pokemon Go.

Milestone 13

Embed Google Analytics in your application and give us a screenshot of the report. Make sure you embed the tracker at least 48 hours before submission deadline as updates are reported once per day.

Home

Analytics All accounts > ZeroSum All Web Site Data ▾

HOME CUSTOMISATION Reports REAL-TIME AUDIENCE ACQUISITION BEHAVIOUR CONVERSIONS DISCOVER ADMIN

Google Analytics Home

Users Sessions Bounce Rate Session Duration

0 0 0 0

21 Sep 22 23 24 25 26 27

Last 7 days ▾ AUDIENCE OVERVIEW ▶

INSIGHTS

Active Users right now 1

Page views per minute

Top Active Pages /Games Feed Active Users 1

REAL-TIME REPORT ▶

How do you acquire users?

Traffic Channel Source/Medium Referrals

21 Sep 22 23 24 25 26 27

Direct Other

Last 7 days ▾ ACQUISITION REPORT ▶

This screenshot shows the Google Analytics Home dashboard. It includes a summary card with user, session, bounce rate, and session duration metrics, followed by a line chart showing user growth over the last seven days. A sidebar on the left provides navigation links for various reports. To the right, there's a 'REAL-TIME' section with a bar chart of active users and a 'ACQUISITION' report showing traffic channel data.

Analytics All accounts > ZeroSum All Web Site Data ▾

HOME CUSTOMISATION Reports REAL-TIME AUDIENCE ACQUISITION BEHAVIOUR CONVERSIONS DISCOVER ADMIN

How do you acquire users?

Traffic Channel Source/Medium Referrals

21 Sep 22 23 24 25 26 27

Direct Other

Last 7 days ▾ ACQUISITION REPORT ▶

How are your active users trending over time? How well do you retain users?

Active Users User retention

200 Monthly Week 0 Week 1 Week 2 Week 3 Week 4 Week 5

This screenshot shows the Google Analytics Acquisition report and User retention report. The Acquisition report displays a bar chart of traffic from different channels over the last seven days. The User retention report shows a line chart of active users over five weeks, with a monthly breakdown.

Analytics All accounts > ZeroSum All Web Site Data ▾

Search reports and help

HOME

CUSTOMISATION

Reports

- REAL-TIME
- AUDIENCE
- ACQUISITION
- BEHAVIOUR
- CONVERSIONS

DISCOVER

ADMIN

Last 7 days ▾

How are your active users trending over time?

Active Users

Date Range	Monthly	Weekly	Daily
29 Aug - 05 Sep	~146	~146	~52
05 Sep - 13 Sep	~146	~146	~52
13 Sep - 20 Sep	~146	~146	~52
20 Sep - 27 Sep	~146	~146	~200

ACTIVE USERS REPORT >

How well do you retain users?

User retention

	Week 0	Week 1	Week 2	Week 3	Week 4	Week 5
All Users	0%	0%	0%	0%	0%	0%
12 Aug - 18 Aug	0%	0%	0%	0%	0%	0%
19 Aug - 25 Aug	0%	0%	0%	0%	0%	0%
26 Aug - 1 Sep	0%	0%	0%	0%	0%	0%
2 Sep - 8 Sep	0%	0%	0%	0%	0%	0%
9 Sep - 15 Sep	0%	0%	0%	0%	0%	0%
16 Sep - 22 Sep	0%	0%	0%	0%	0%	0%

Last 30 days ▾

ACTIVE USERS REPORT >

Last 6 weeks ▾

COHORT ANALYSIS REPORT >

When do your users visit?

Users by time of day

Where are your users?

Sessions by country

What are your top devices?

Device	Percentage
Desktop	75.7%
Mobile	24.3%

Analytics All accounts > ZeroSum All Web Site Data ▾

Search reports and help

HOME

CUSTOMISATION

Reports

- REAL-TIME
- AUDIENCE
- ACQUISITION
- BEHAVIOUR
- CONVERSIONS

DISCOVER

ADMIN

Last 30 days ▾

When do your users visit?

Users by time of day

Where are your users?

Sessions by country

What are your top devices?

Sessions by device

Device	Percentage
Desktop	75.7%
Mobile	24.3%

Last 6 weeks ▾

COHORT ANALYSIS REPORT >

Last 30 days ▾

ACTIVE USERS REPORT >

Last 7 days ▾

LOCATION OVERVIEW >

Last 7 days ▾

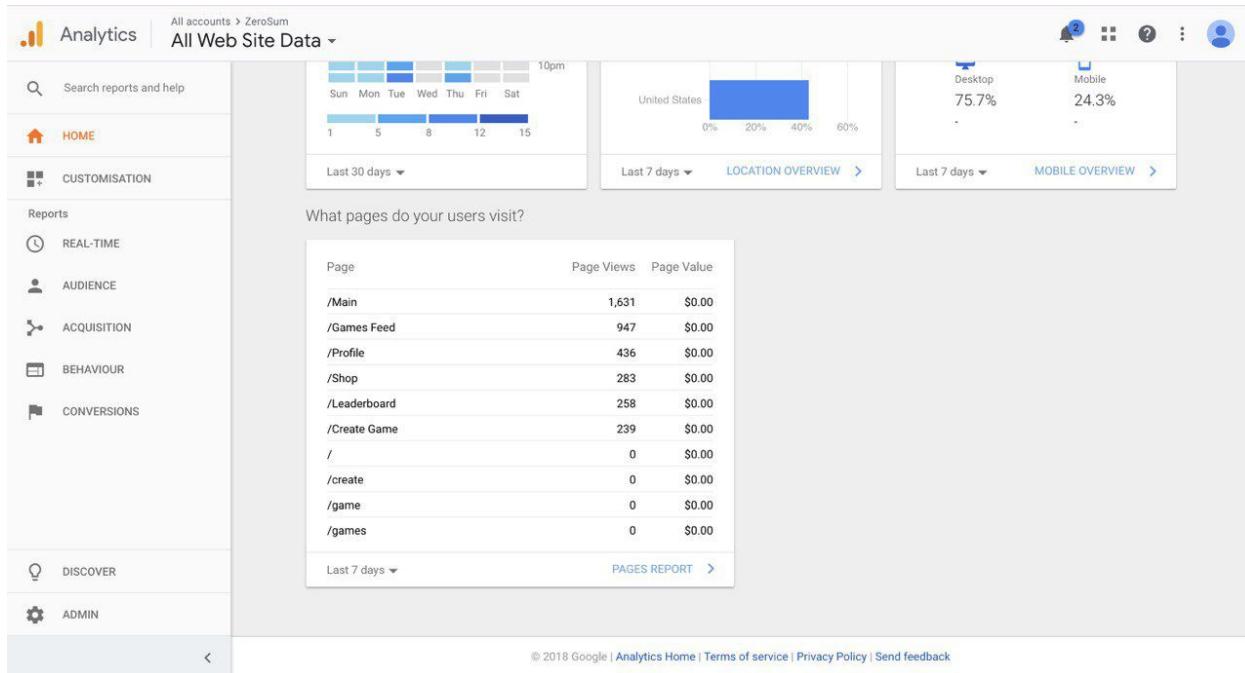
MOBILE OVERVIEW >

What pages do your users visit?

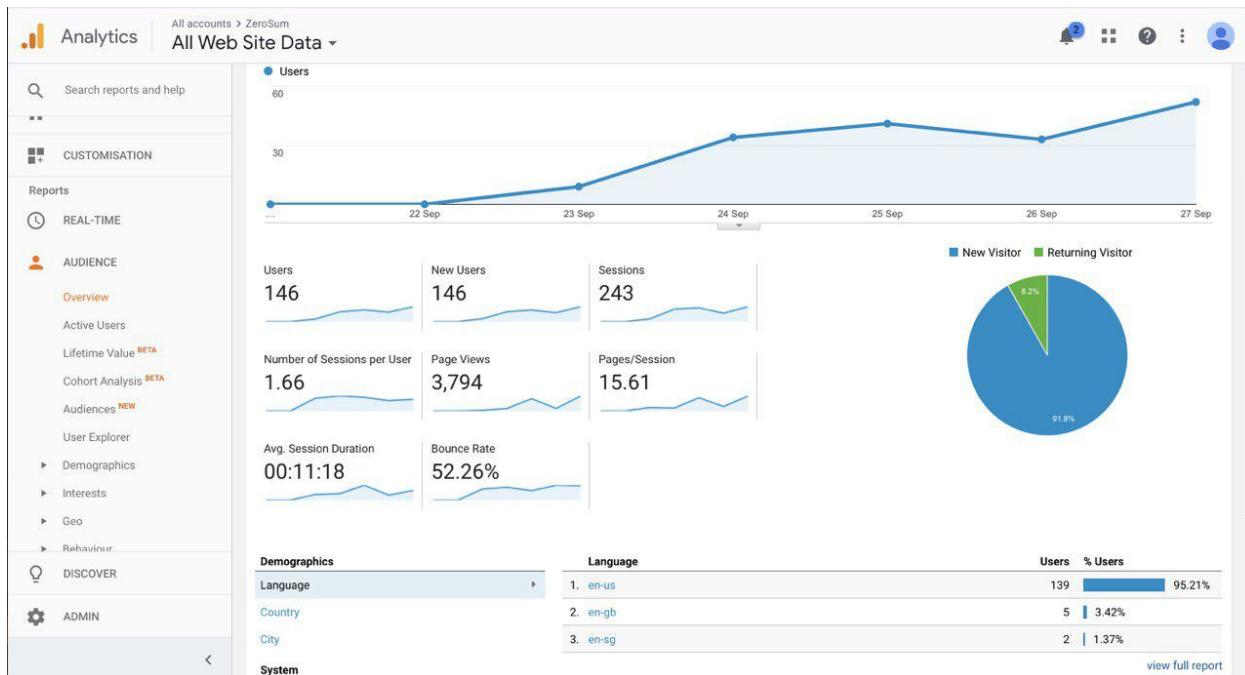
Page

Page Views

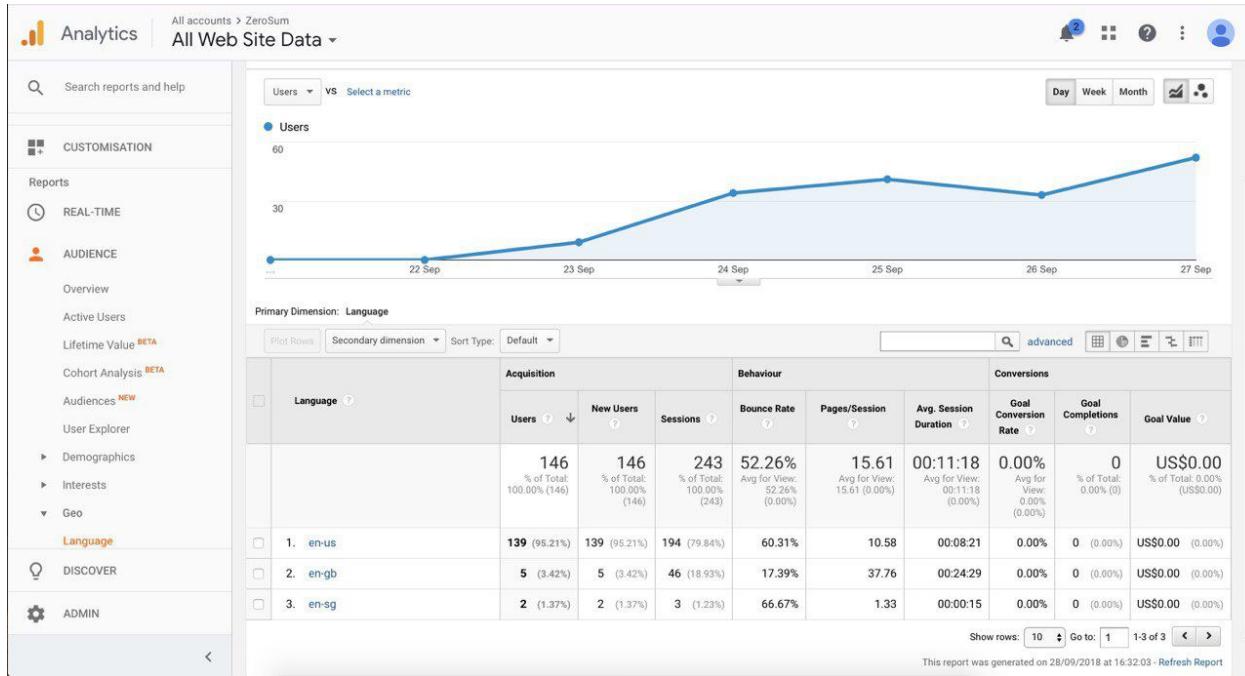
Page Value



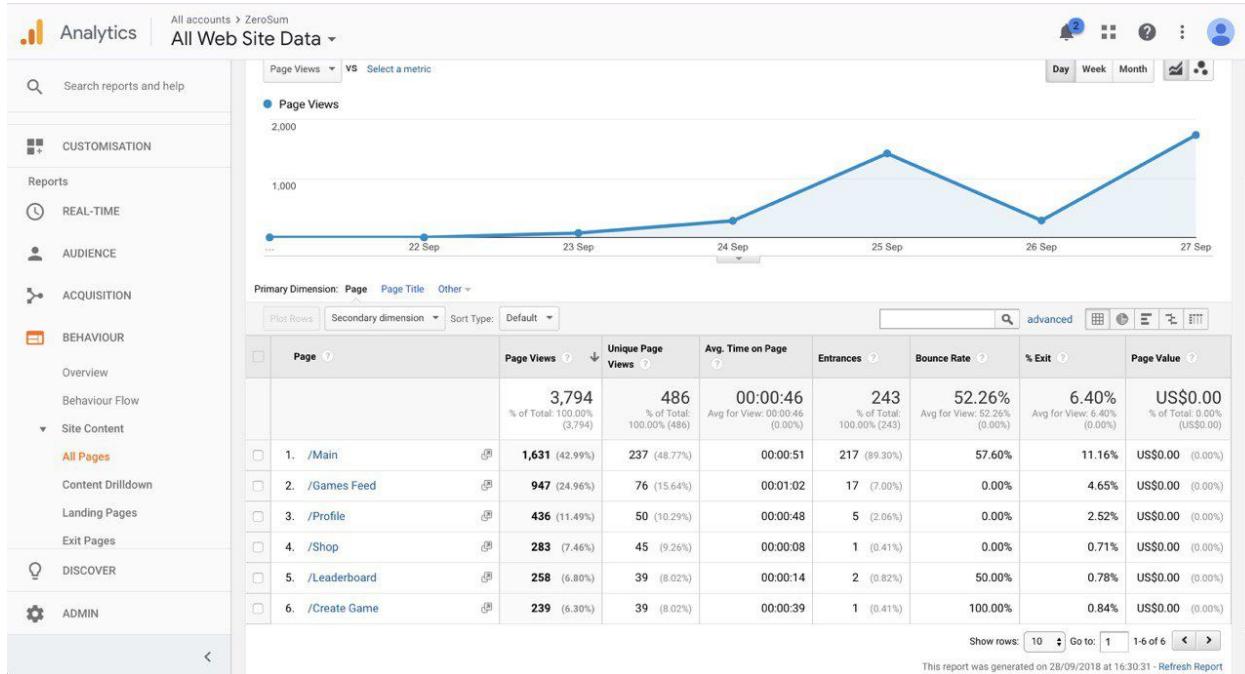
Audience Overview



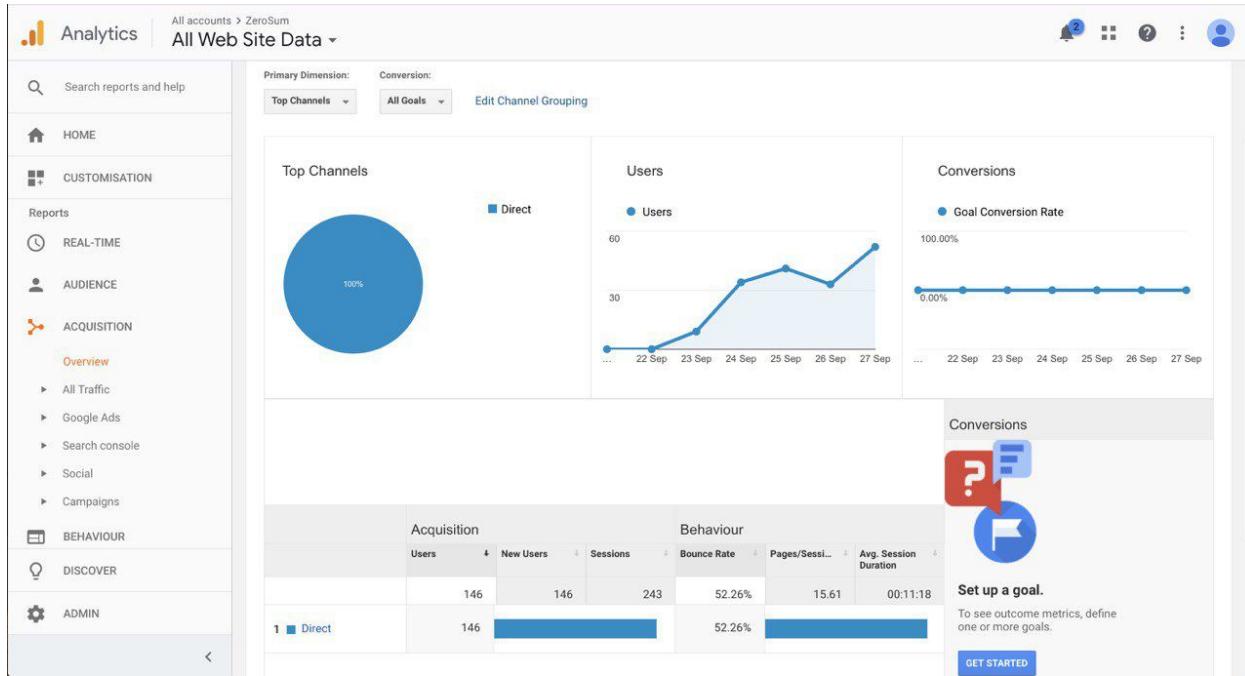
Audience



Behaviour



Acquisition Overview



Milestone 14

Achieve a score of at least 90 for the Progressive Web App category and include the Lighthouse html report in your repository.

Milestone 15

Identify and integrate with social network(s) containing users in your target audience. State the social plugins you have used. Explain your choice of social network(s) and plugins. (Optional)

The social plugin that ZeroSum has integrated is Facebook. There are a couple of reasons of why we have decided to use Facebook:

1. Creating competition amongst friends

Since this is a gamified platform with statistics for the users, users are able to compare their win rates with each other, creating the element of competitiveness as the game aesthetics. In this case, the aesthetics we are talking about himi is not the appearance of the game, but it's according to the game design MDA framework. (Mechanics, Dynamics, Aesthetics). The Aesthetics in the MDA framework is talking about the feeling invoked from the users. Hence, we link back the feeling of competitiveness to the aesthetics of the game.

2. Playing games your friends have created

Playing games with friends will be more fun as compared to just playing alone and strangers, as there are times that users would want to play what their friends have created and arguing with friends what are the better choices to make and even discuss more about the question. Hence, it will be more interesting to integrate friends system in our application as well.

3. More convenience to the users

We provided the Facebook as the platform to sync the users' profile to because most of the applications and games hosted on the application store offers Facebook as an account to login to and sync your process. We are also able to utilise their existing Facebook profile to generate their profile on our application. Thus, from users' point of view, it will be much more convenient for them if they don't create an extra account for our application, but using an **existing** platform out there to make things easier and centralized for them.