Documentation PAS

Sergio Manuel González Vargas - A01745446

Rodrigo Alfredo Mendoza España - A01720627

Gala Flores García - A01745037

Diego Yunoe Sierra Díaz - A01752219

Antonio Oviedo Paredes - A01752114

Subject:

Construcción de software y toma de decisiones

Group: 401

April 2022

Instituto de Estudios Superiores del Tecnológico de Monterrey

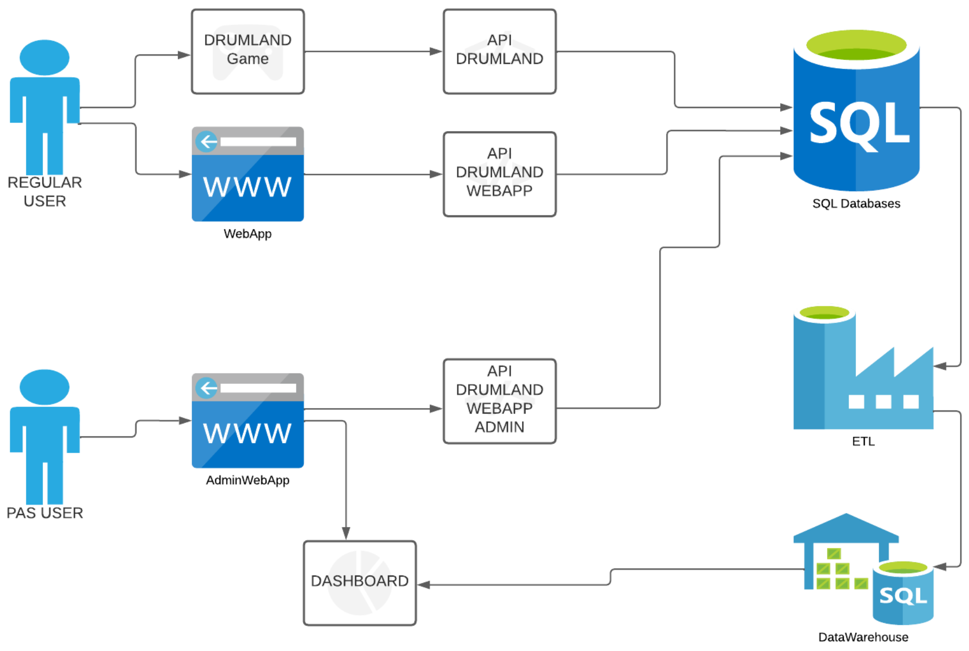
1. Executive presentation of the proposed solution.

https://www.canva.com/design/DAE_hUrcUok/zF9X571rtZ6dOV6Re2osyA/view?utm_content=DAE_hUrcUok&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

2. Functional specification and system architecture.

- Unit: 2021.2.16f8
- 3 complete levels 2d
- available on Mac OS (Intel, M1), windows, linux.
- genres: Interactive stories and Drumming simulators.
- focused on Mental relaxation through percussion instruments.
- provide private information (user top score, gender, age, country, etc.)
- provide public information (top score, country, etc.)
- provide public information (top score, country, etc.)
- The game is designed for a single player

System Architecture



Drumland

Drumland is our game created in Unity Game Engine, which is in charge of gathering all the resources to build a videogame.

In a hybrid 16 bit/8 bit world, Drumland is an adventure/platform/retro/music game themed, the game tells the story about DRUMLAND, a magic altern universe where Jorge Luis (NinjaFrog) will face the evil wizard, whose goal is to keep the percussion power of music.

App Services

Azure App Services allows users to quickly and easily create enterprise-ready web and mobile apps for any platform or device, and deploy them on a scalable and reliable cloud infrastructure. This is to create, build, deploy, and scale web apps and APIs with the programming language of our choice, which we chose PHP.

We used Azure App Services to host our PHP files. These PHP files allow the game to connect directly to the transactional database. Using code, it is able to register a user to the database which registers: Username, password, birthdate, country, sex, first name and last name. This is permanently registered as a new record, so that a user can immediately log in with the new information. It is also possible to log in using the username and password. If those records already exist on the database with the correct data, a user is able to successfully log in and play.

Using app services and unity code, every time a user logs in and logs out, it is registered to the database with the starting time and finish time of that session. Also, every time a user plays a level it is registered, this to maintain a record of every player's playtime.

The app service used is located in the region of US East, since it was the closest to our work environment. Using a runtime stack of PHP 7.4 with Windows as its operating system. Its SKU is Free F1 which is free forever, however it has a shared infrastructure with only 1 GB of memory.

Static Web Apps

Azure Static Web Apps allows developers to host static content with the ability to integrate serverless APIs.

We used this service to deploy our web apps by creating an account in Azure
and simply connecting our github repository to azure services.

The azure static web apps let us develop in almost any last generation framework meaning that it will not matter where you want to build your app.

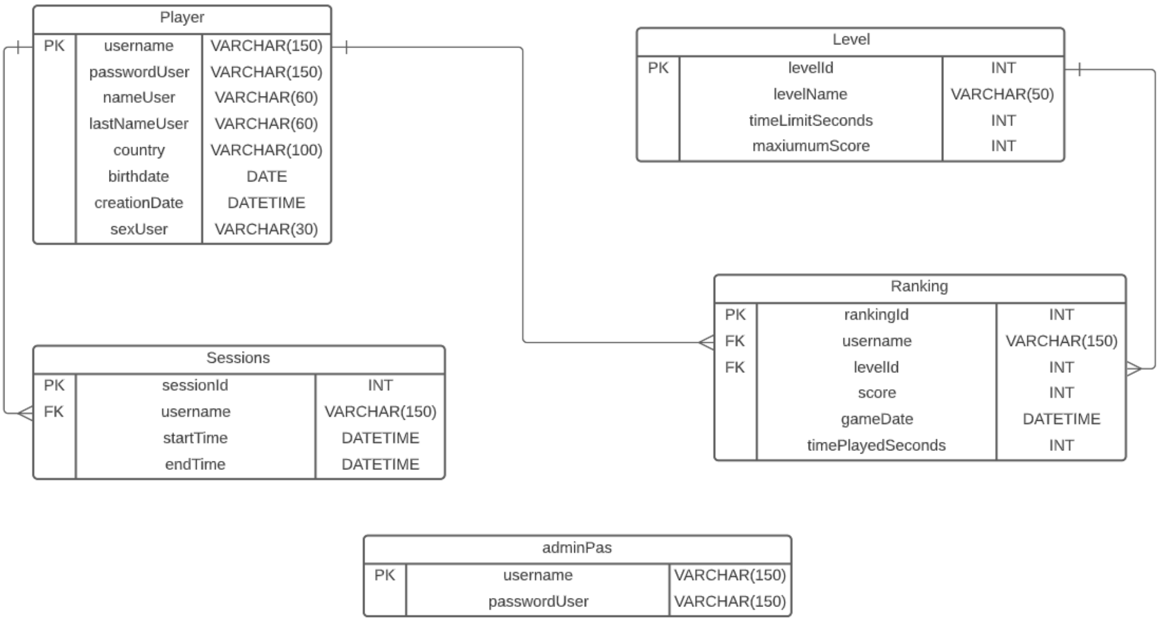
Finally it does not matter if the project is a personal project or a specific development since azure let us pay for what we need.

Transactional Database

We use a transactional database, where we store all of our data of the video game such as users and their information, level information, sessions and rankings. Using a relation model for our database, which means that it is a collection of data items with pre-defined relationships between them. These items are organized as a set of tables with columns and rows. Such as our app services, we use the service Azure provides to host them.

Azure SQL Databases is an always-up-to-date relational database service built for the cloud. It automatically scales to meet your apps' requirements and keeps them running with up to 99.995% availability.

The relational model for the database is the following:



Player		
PK	username	VARCHAR(150)
	passwordUser	VARCHAR(150)
	nameUser	VARCHAR(60)
	lastNameUser	VARCHAR(60)
	country	VARCHAR(100)
	birthdate	DATE
	creationDate	DATETIME
	sexUser	VARCHAR(30)

Level		
PK	levelId	INT
	levelName	VARCHAR(50)
	timeLimitSeconds	INT
	maxiumumScore	INT

Ranking		
PK	rankingId	INT
FK	username	VARCHAR(150)
FK	levelId	INT
	score	INT
	gameDate	DATETIME
	timePlayedSeconds	INT

Sessions		
PK	sessionId	INT
FK	username	VARCHAR(150)
	startTime	DATETIME
	endTime	DATETIME

adminPas		
PK	username	VARCHAR(150)
	passwordUser	VARCHAR(150)

This means that for every player that registers through the video game, the database will store the username, password, first name, last name, country of origin, birthdate, the day the user was created and the sex. The information of each level such as the name, the time limit and maximum store are also stored. Whenever a user logs in and logs out, the time when they log in and the time when they log out are stored. Whenever a user plays a level, the whole information for that level is stored such as who played that level, which level they played, how long it took them and the score. There is also a place where the login information for the admins who will be able to see the dashboards in the web page.

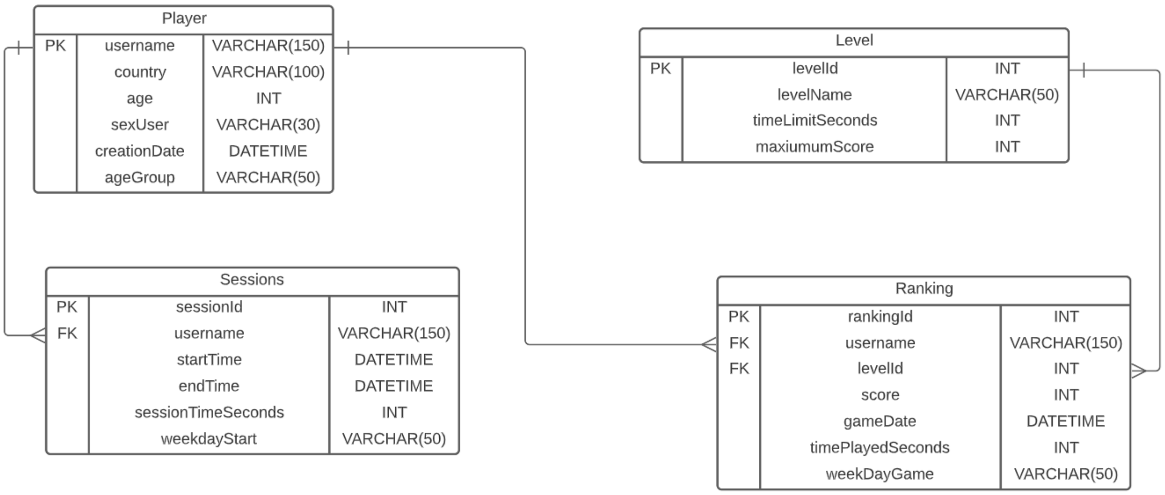
The database is located on the region of East US using 10 DTU's with a maximum size of 250 GB. For this database, the approximate cost per month is 15 USD.

The server name to access the database is the following: drumland.database.windows.net
Credentials are not shown here due to security reasons.
Also, worth mentioning that the IP address of the service/user that is trying to access the data must be added to the server firewall.

Analytical Database (Data warehouse)

For the analytics database, we use a very similar format, such that it is a relational database and that it is hosted on Azure SQL Database service. The difference is that we use this one particularly for analytical purpose where we will graph and show the data through dashboards, which is explained further below. The transactional database automatically transfers its data whenever a new record appears with modified columns to adjust to our analytical needs.

The relational model for the analytical database is the following:



Player

PK	username	VARCHAR(150)
	country	VARCHAR(100)
	age	INT
	sexUser	VARCHAR(30)
	creationDate	DATETIME
	ageGroup	VARCHAR(50)

Level

PK	levelId	INT
	levelName	VARCHAR(50)
	timeLimitSeconds	INT
	maxiumumScore	INT

Sessions

PK	sessionId	INT
FK	username	VARCHAR(150)
	startTime	DATETIME
	endTime	DATETIME
	sessionTimeSeconds	INT
	weekdayStart	VARCHAR(50)

Ranking

PK	rankingId	INT
FK	username	VARCHAR(150)
FK	levelId	INT
	score	INT
	gameDate	DATETIME
	timePlayedSeconds	INT
	weekDayGame	VARCHAR(50)

As it is seen, the password and names of the users are irrelevant, the birthdate is replaced with the age of the user, and we added an age group to filter our data. We also added the duration time in seconds per session, as well as which day of the week they logged on. On ranking, we added the day of the week which they played.

The database is located on the region of East US using 10 DTU's with a maximum size of 250 GB. For this database, the approximate cost per month is 15 USD.

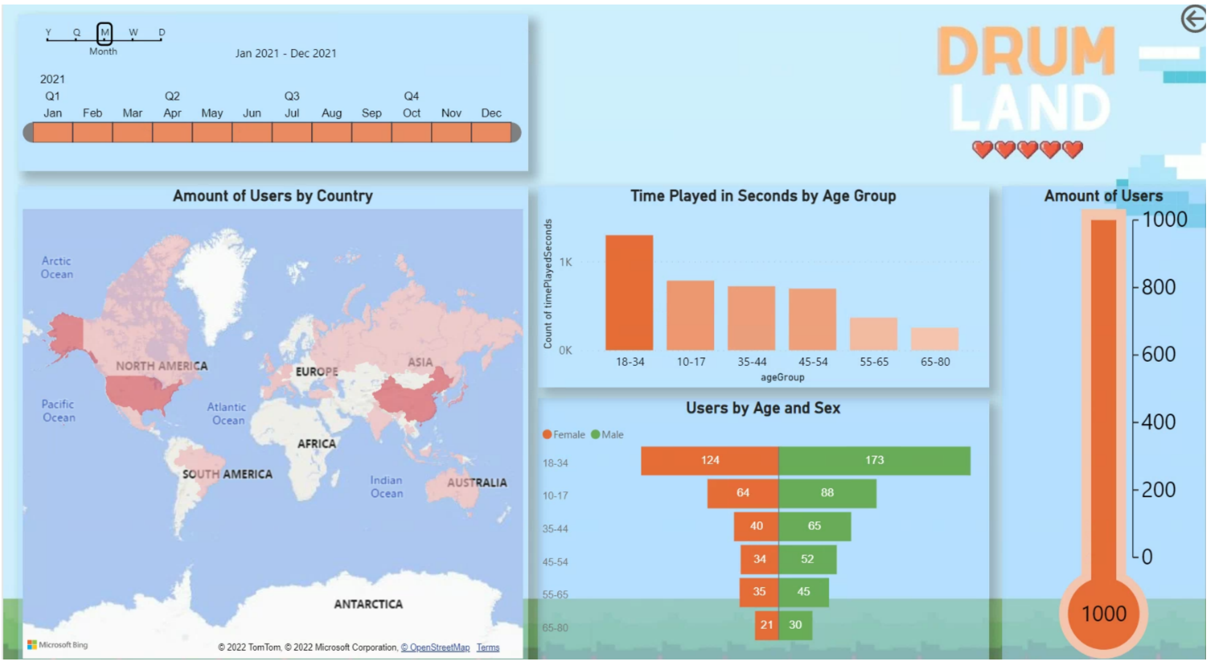
The server name to access the database is the following: drumland.database.windows.net
Credentials are not shown here due to security reasons.

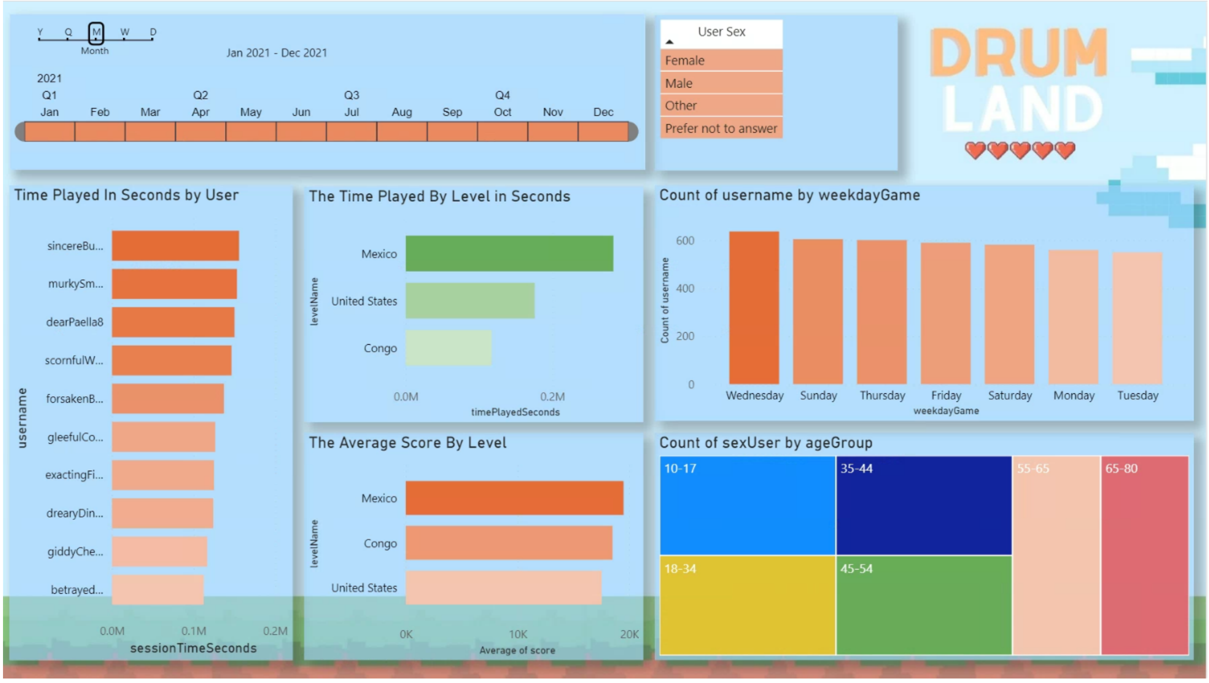
Also, worth mentioning that the IP address of the service/user that is trying to access the data must be added to the server firewall.

DASHBOARD

Using the analytical database, a dashboard was created. The one created right now is using Power BI, but any other visualization tool works as well. To connect using the visualization tool, the server name: drumland.database.windows.net using the database: drumlland-DW with the necessary credentials is required.

The current dashboard is composed of two pages:





To access the public dashboard, use the following URL:
https://app.powerbi.com/view?r=eyJrIjoiYmQyN2Q4ODItY2E5OC00ZWM5LTgyNjQtYzIwYjRlZWUwNTc1IiwidCI6ImM2NWEzZWE2LTBmN2MtNDAwYi04OTM0LTVhNmRjMTcwNTY0NSIsImMiOjR9&pageName=ReportSection63de8cf0b9aacd1119b6

The data in this dashboard is hypothetical and computer generated, however when there are real users in the video game, the same analytical method will be used.

3. Summary of amount of effort (hours) applied in the inception, design, construction, testing and release of the system.

	Hours	**Cost**	**Total**
Inception	5	**$70 / h**	**$350 USD**
Design	**10**	**$70 / h**	**$700 USD**
Construction	**275**	**$70 / h**	**$19250 USD**
Testing	**35**	**$70 / h**	**$2450 USD**
Release of the	**4**	**$70 / h**	**$280 USD**

system			

4. Videos where the project features are demonstrated, showing the information flows for each user type. Delivery: Open link to youtube. 10 minutes or less in length.

Video

https://youtu.be/M3pQL-xMZN4