TEC | Tecnológico de Costa Rica

Instituto Tecnológico de Costa Rica

Computer Engineering School

Professor: Marco Rivera Meneses

Course: Bases de datos – CE3101

Group 1

Participants:

Emanuel Marín Gutiérrez – 2019067500

Jose Andrés Rodríguez Rojas – 2019279722

Oscar Soto Varela – 2020092336

Sebastián Chen Cerdas – 2021571438

Project II – StraviaTEC

Technical documentation

II Semester, 2023

Table of Contents

Source code

GitHub repository link: [StraviaTEC - GitHub Repository]

Conceptual model

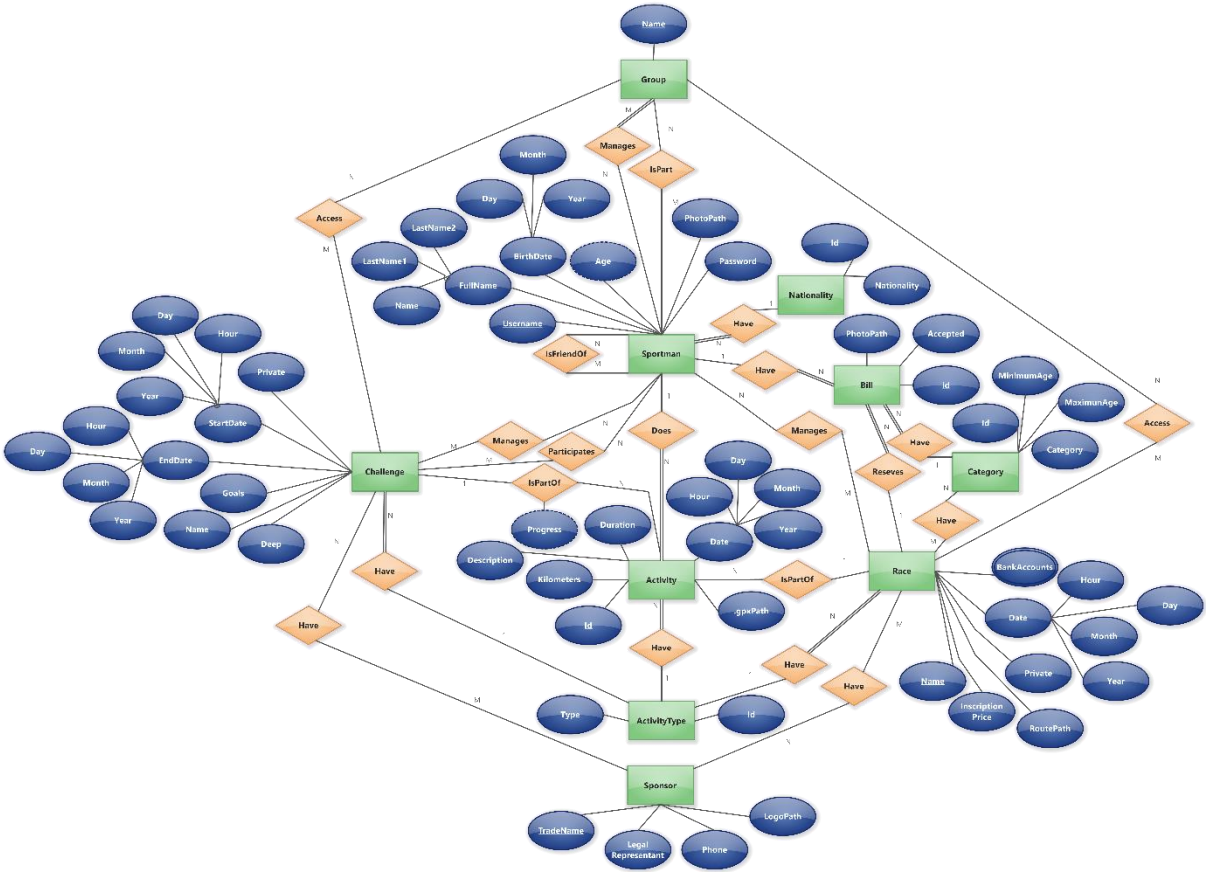


Figure 1. Conceptual model

Relational model

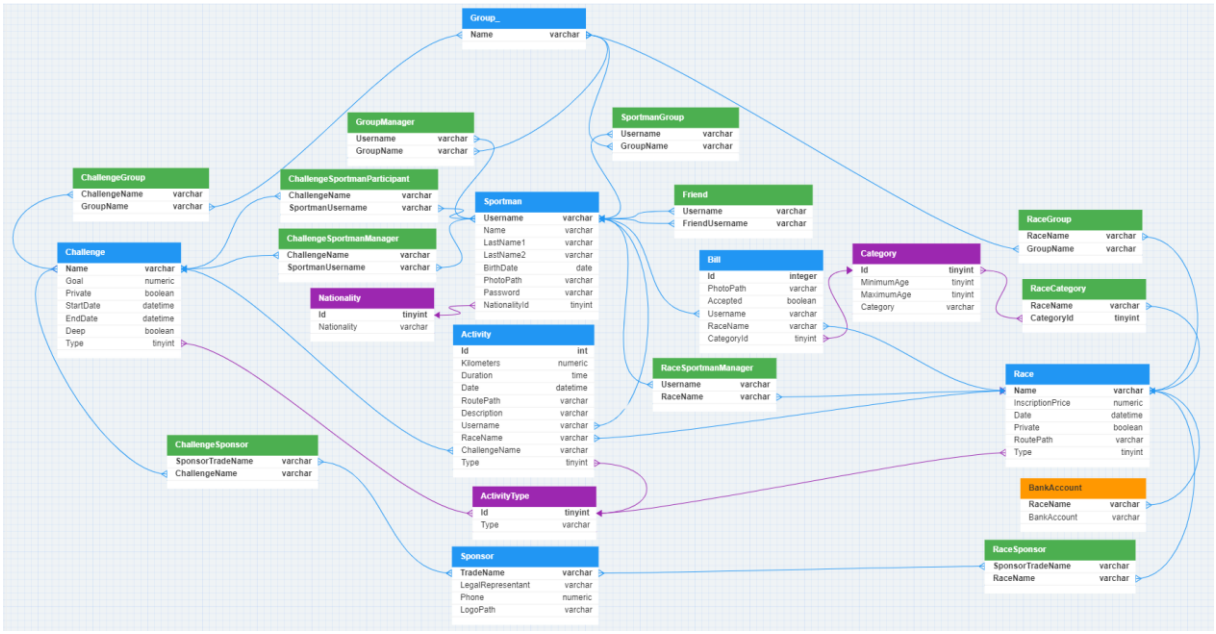


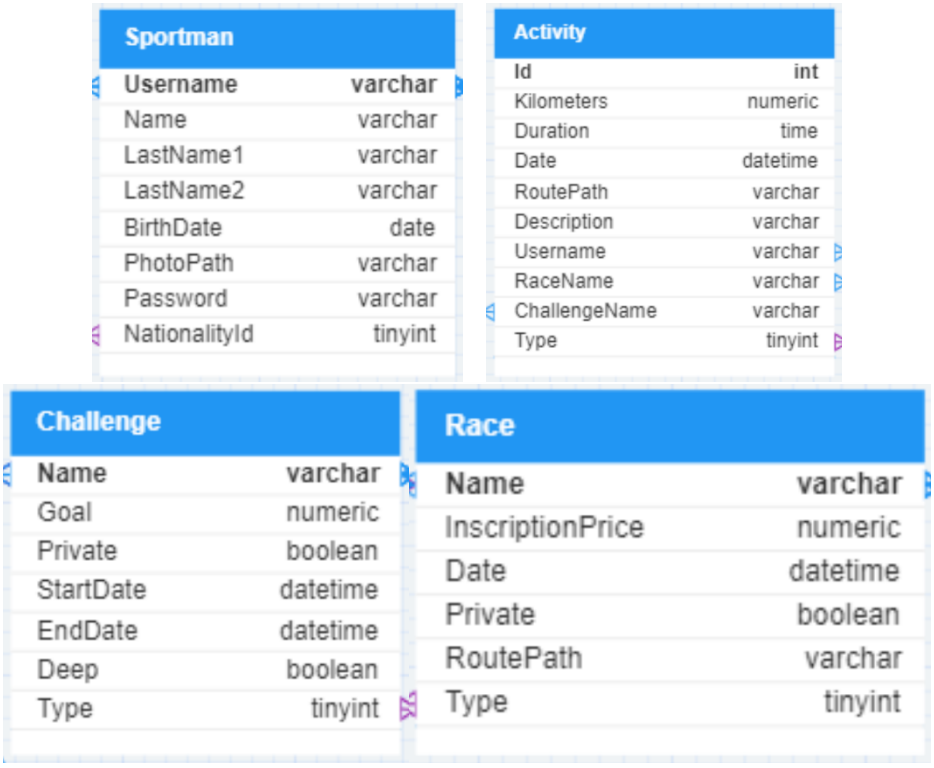
Figure 2. Relational model

Mapping steps:

For the following mapping steps consider the conceptual model presented on the last section of this document (notice that every relation on the model is binary, there are no complex relations).

Strong Entity Maping:

For each strong entity there was created a table using the simple attributes of the strong entity, here you can find examples of some of those strong entities:



Weak Entity Maping:

For each weak entity there was created a table using its simple attributes and, there must be added the primary attribute of the related entities. For this project, there were not weak entities, due to that lack of weak entities, there are no examples of weak entities mapping.

1:1 Entity Relation Maping:

In terms of 1:1 relations mapping, there are three possible scenarios:

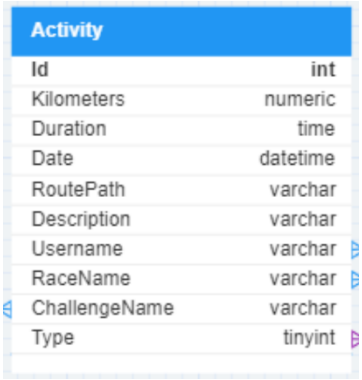
a. Create a foreign key attribute on the table with total participation among the relation.
b. Create a double foreign key reference on every table.
c. Creating an extra table, considering every table's primary key as a foreign key and including relation attributes (In case there exists any relation attribute).

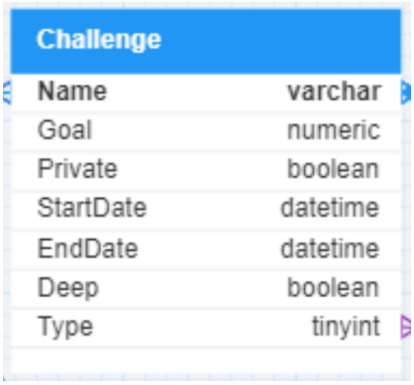
In the same way as weak entities, this project doesn't require of this kind of relations, that means there are no examples for this mapping method.

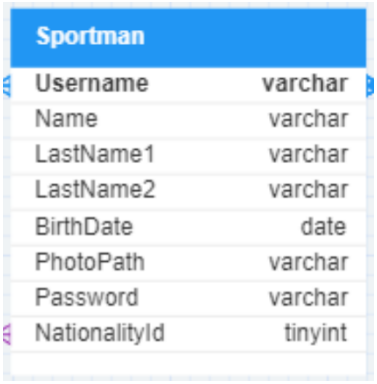
N:1 Entity Relation Maping:

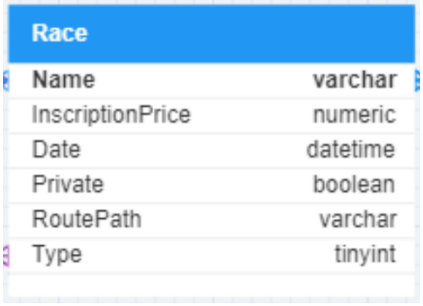
For N:1 relations mapping, there must be created a foreign key on the table that has participation N among the relation, as it is shown on the following examples:

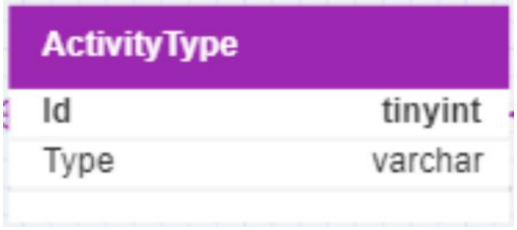
The entity Activity have 1:N relationships with entities Sportman, Challenge, Race and ActivityType, because of that it includes “Username”, “RaceName”, “ChallengeName” and “Type” attributes, that refrences the primary attributes of each relations’ tables



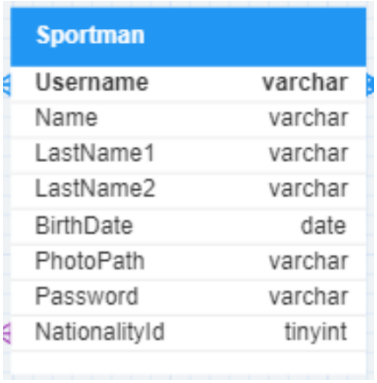








As well, the entity Sportman has N:1 relationship with Nationality; Sportman has “NationalityId” attribute due to the N participation on Sportman side of relationship.



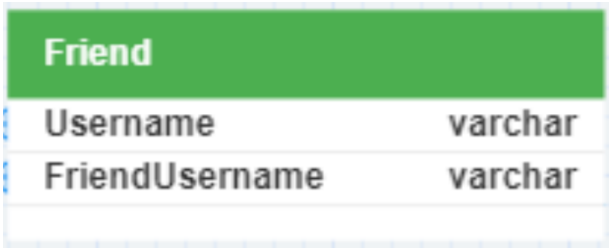
N:N Entity Relation Maping:

For the N:M relationships, there must be created an extra table, considering as a primary key, the combination of every participant tables' primary keys (those table attributes represent the primary key and foreign keys of the new table). Following are some examples of those relationships mapping technique:

Race (Primary key: Name) and Sponsor (Primary key: TradeName), have N:M realtionships, so the RaceSponsor table is created with SponsorTradeName and RaceName attributes as primary key (together) and foreign keys (each one).

Challenge (Primary key: Name) and Sponsor (Primary key: TradeName), have N:M realtionships, so the ChallengeSponsor table is created with SponsorTradeName and ChallengeName attributes as primary key (together) and foreign keys (each one).

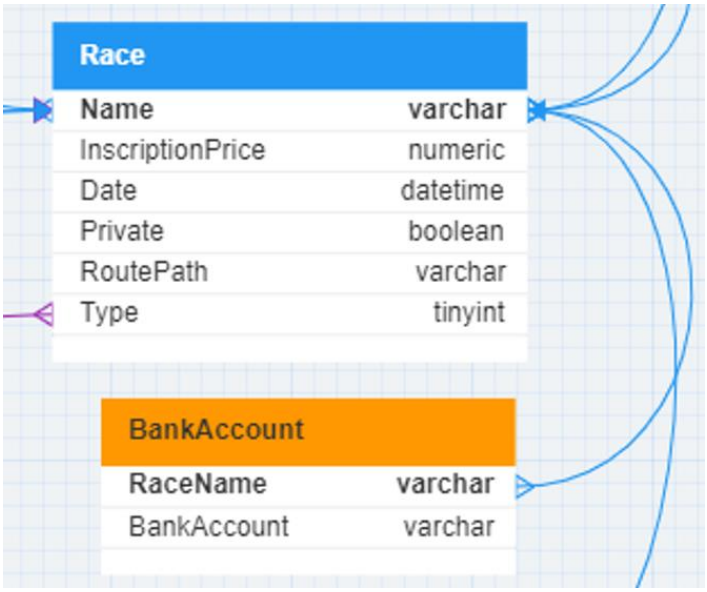
Sportman (Primary key: Username) and itself, have N:M realtionships, so the Friend table is created with Username and FriendUsername attributes as primary key (together) and foreign keys (each one).



Race (Primary key: Name) and Category (Primary key: Id), have N:M realtionships, so the RaceCategory table is created with RaceName and CategoryId attributes as primary key (together) and foreign keys (each one).

Normal Forms

It was ensured compliance with all the five normalization forms in the design and creation of the database, you can check it out in the database Relational Model. But here is made an emphasis on the compliance of the fourth normal form 4FN in the design of a multivalued attribute BankAccount in the database Relational Model. As it can be seen in the figure below it respects both the 3FN (Transitive dependence) because the relation doesn't have nonkey attributes functionally determined by another nonkey attribute. And also respects the 4FN by applying a separate table for the multivalued attribute BankAccount of the table Race thanks to the correct compliance of the mapping steps from a well-designed Conceptual Model to make the Relational Model.



Data structures developed

Sportman

This table is used to store all sportmen basic personal information

Activity

This table stores the information related to an activity, such as the activity date, the duration of the session, if it is part of a challenge or a race, and other information.

Challenge

This table saves the challenge basic information, like its name, date interval to be available, goal, privacy and more important information.

Race

This table is used to keep the race basic information: name, date, privacy, etc (Similar to challenges)

Sponsor

This table stores the sponsors available in the app and its information, like the trade name, commercial name, representant, phone number, etc.

Bill

This table is meant to keep all races inscriptions requests and its information (user interested to join the race, category to join, receipt picture, etc). When any user tries to join a race, a bill is posted on this table, waiting for an approval from the race creator.

Group

This table saves the group basic information (name).

Nationality

This table works as a dictionary of nationalities, so it is used to keep a nationality related to an id (to allow a better performance on sportman table).

Category

This table works as a dictionary of categories, so it is used to keep a category related to an id (to allow a better performance on race, challenges and other tables).

ActivityType

This table works as a dictionary of activity types, so it is used to keep a type of activity related to an id (to allow a better performance on race, challenges and other tables).

Friend

This table is used to save all friendships among the app users, every tuple of this table relates a user (using its username) to another user (using its username as well).

ChallengeSportmanParticipant and ChallengeSportmanManager

These tables are two middle tables with different usages, ChallengeSportmanParticipant stores all users submitted to a challenge (relates a user by its username to a challenge by its name). The ChallengeSportmanManager keeps all users that manage a challenge, it's the same structure as ChallengeSportmanParticipant table but the usage of every table is different.

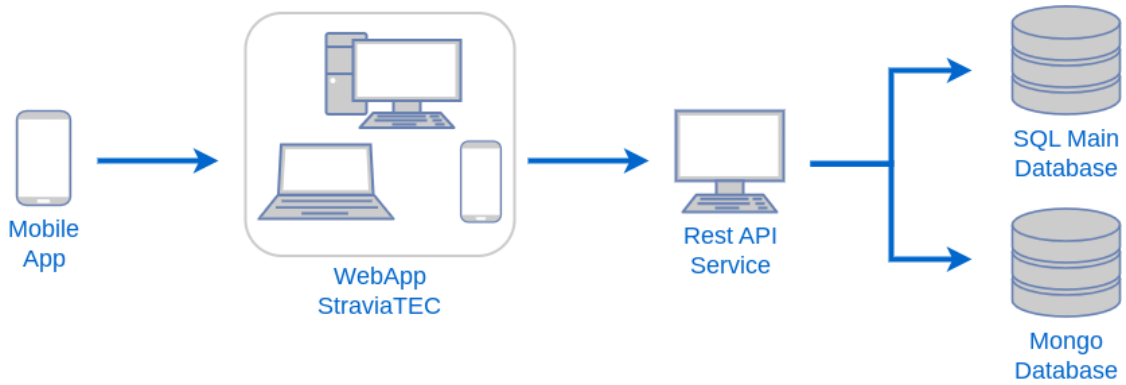
RaceSportmanManager

Like ChallengeSportmanManager table, this table relates a user and a race (using user's username and race's name) and is meant to save all users that manage a race.

ChallengeGroup, ChallengeSponsor, GroupManager, SportmanGroup, RaceGroup, RaceCategory and RaceSponsor

These tables are standard middle tables, every of them has descriptive names and only store the primary keys of each table related by them (In order to relate and keep relationships between two tables).

Developed architecture



The mobile app registers a map trace performed by the user and generates a GPX file, then the user interacts with the web app (using or not the GPX file generated) and requests the API service to get and post information to databases; commentaries are stored on the mongo database and every other information goes to SQL main database.

Found problems

GPX file load and display on StraviaTEC WebApp

Once the GPX file was already generated on the mobile app, there was a trouble finding an angular module that provided an updated version of Google Maps API that allowed to read and display the GPX file on HTML components, that meant a problem for the project development because several modules with deprecated features and functionalities were tested and none of them helped to solve the GPX displaying issue.

After several tries and tests performed by the team, a module was found that allowed to show the GPX file correctly.

Teamwork evidence

Project goals

As part of the development of the final project, the working team must deliver the following documents and deliverables:

- Work plan.
- Conceptual model of the main database.
- Relational model of the main database.

- Database creation scripts.
- Database population scripts.
- Project class diagram.
- Project architecture diagram.
- Executive progress report I.
- Executive progress report II.
- Technical documentation.
- Installation manual.
- User manual.
- REST API.
- Web application: StraviaTEC (for users and admins).
- Mobile application

Work roles

- DBA: Responsible for approving changes to be made to the database and authorizing access to it.
- DBD: Designs the database, considering the conceptual model and its relationships.
- Front-End: Designs and develops the user interface of the website and mobile application.
- Back-End: Designs and develops the functionality of the REST API to connect the necessary systems.

Rules

1. Present individual progress to the team within intervals of up to 4 days.
2. Maintain active communication with the team through the agreed-upon means.
3. Be punctual for team meetings, and in case of inability to attend, read the meeting minutes and log.
4. Play a game of table football at least once a week.

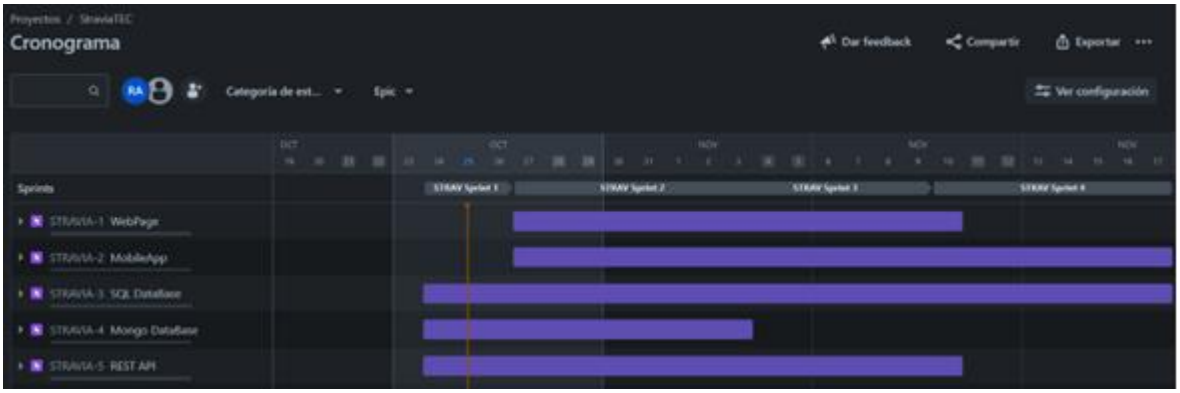
Note: Non-compliance with the rules described above may result in a penalty or even expulsion from the workgroup.

Time schedule and project advancements

For the distribution of workloads and the definition of relevant deadlines for each of the deliverables and tasks in the development of the final prototype, we will be using Jira as the software to track the progress of work and the updates submitted by team members.

Link to the Jira repository: Jira repository: StraviaTEC

Note: To view the task organization and teamwork, access to the Jira repository is required. Below is a view of the initial work schedule accessible in the Jira repository.



Meeting minutes

Meeting #1:

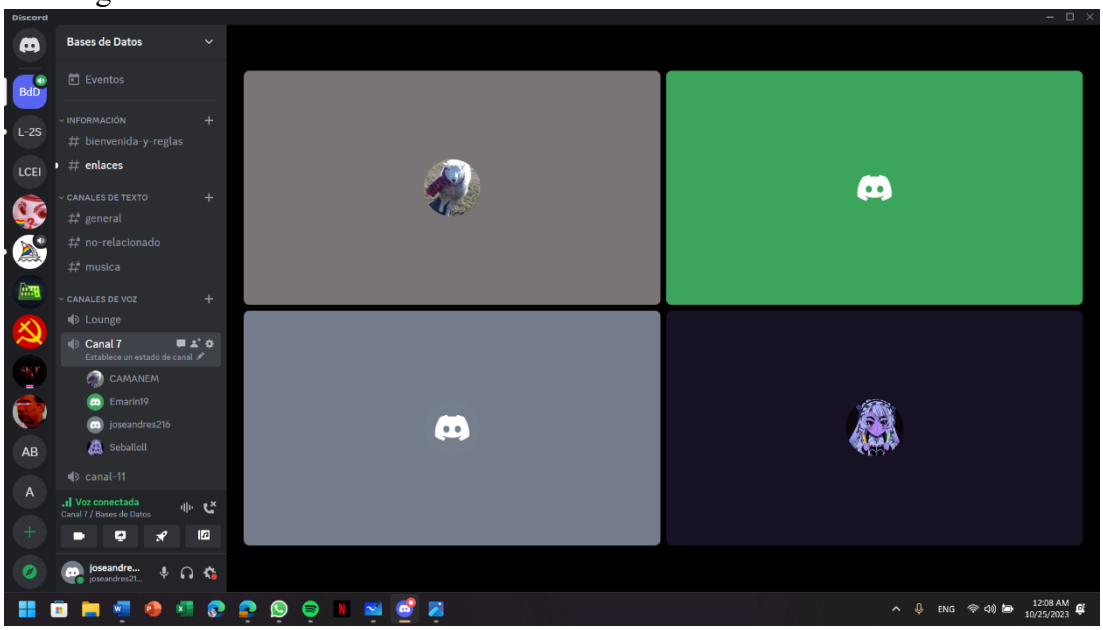
Date and time of the meeting: October 24, 2023 – 8:00 p.m.
Participants: Everyone.
Meeting agenda:

1. Conceptual models review.
2. Relational model mapping.
3. Task creation on Jira repository.
4. Schedule next meeting and tasks.

Minutes of the meeting held:

- The team reviewed all the conceptual models and made a definitive conceptual model for the main database.
- Based on the conceptual model made, the team mapped that model and designed it's relational model.
- There were made all the tasks and it's respective distribution (using Jira sprints) for the whole project development.

Meeting evidence:



Meeting #2:

Date and time of the meeting: October 29, 2023 – 8:00 p.m.

Participants: Everyone.

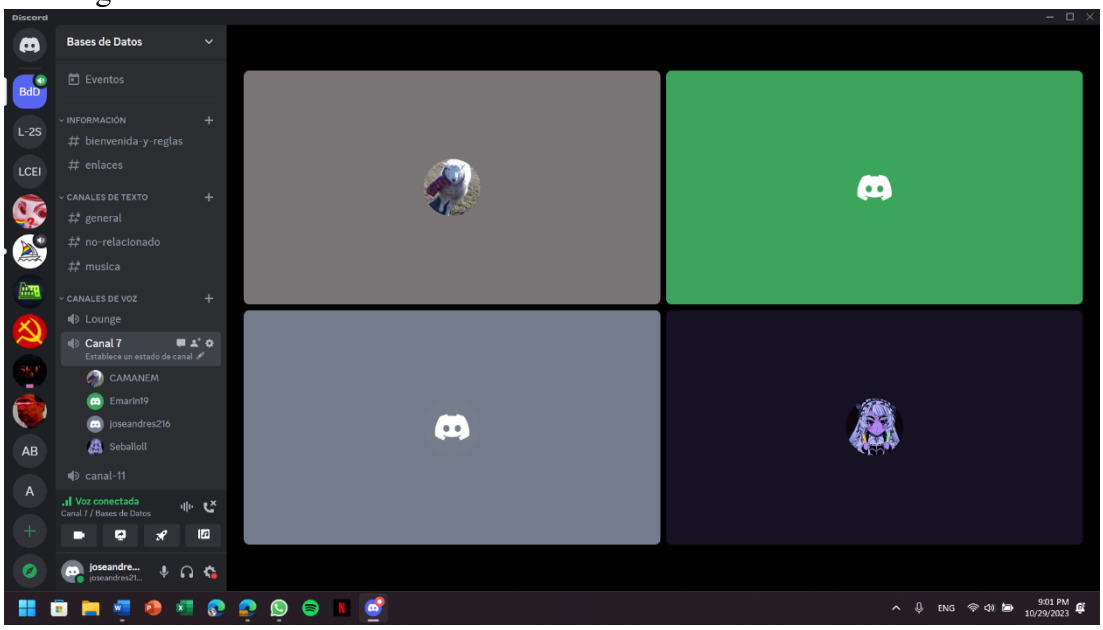
Meeting agenda:

1. Review task for the last week (Update Jira state and sprint)
2. Distribute tasks for the next sprint (and start Jira sprint)

Minutes of the meeting held:

- The team reviewed the tasks completed during the last week.
- The sprint 2 of the Jira repository was started and its tasks properly distributed among the team members

Meeting evidence:



Meeting #3:

Date and time of the meeting: October 31, 2023 – 6:00 p.m.

Participants: Everyone.

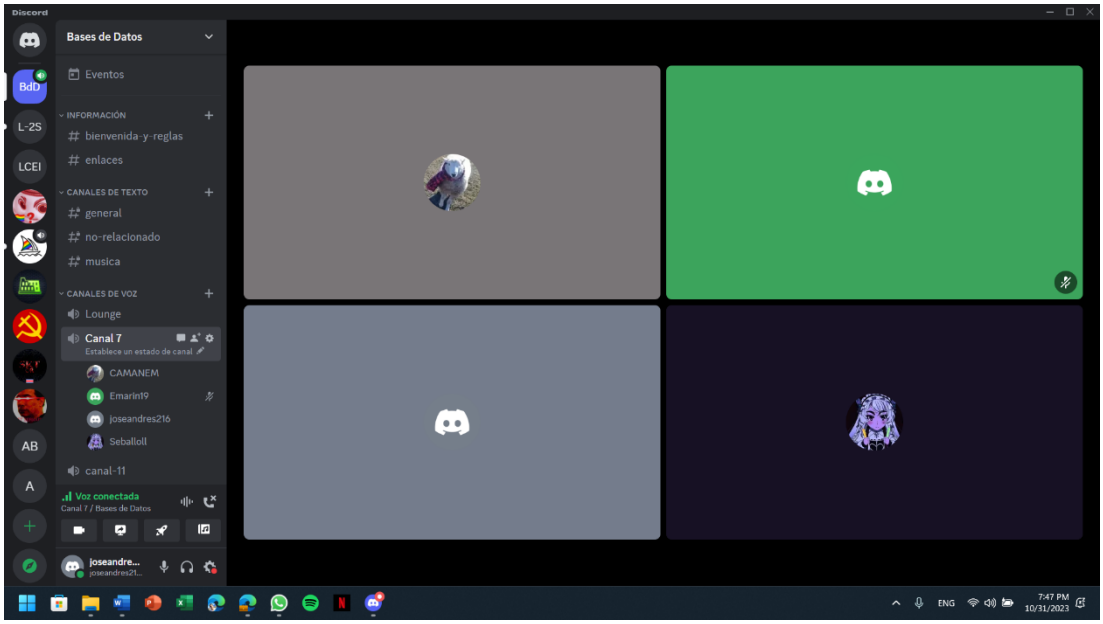
Meeting agenda:

1. Review task for the last week (Update Jira state and sprint)
2. Distribute tasks for the next sprint (and start Jira sprint)

Minutes of the meeting held:

- The team reviewed the tasks completed during the last week.
- The sprint 2 of the Jira repository was started and its tasks properly distributed among the team members

Meeting evidence:



Activities log

Log entry #1:

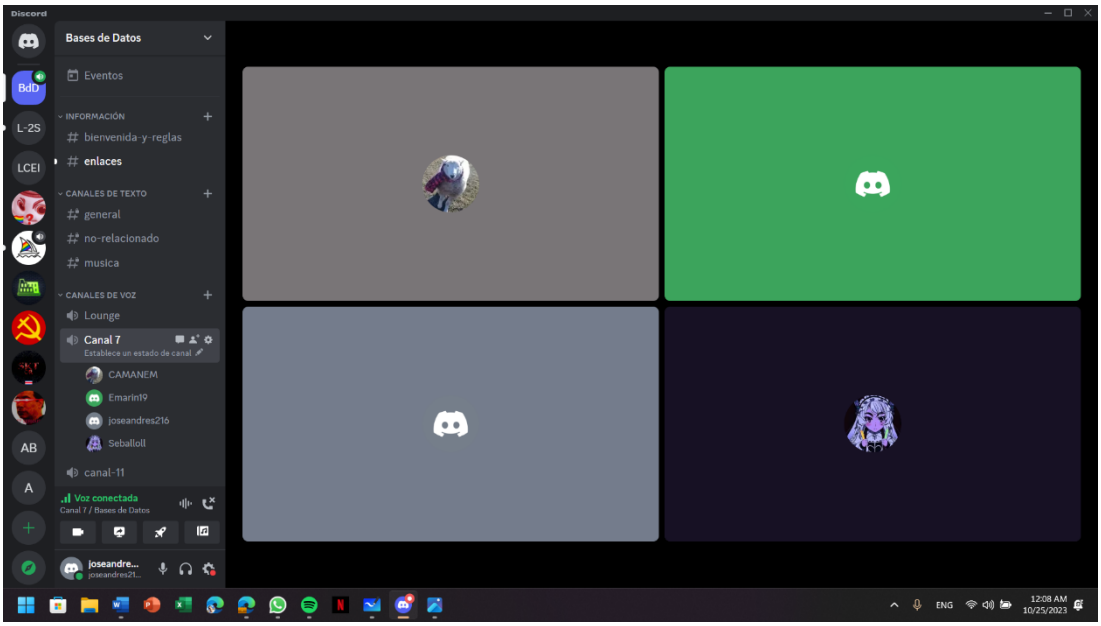
Date and time of the activity: October 24, 2023 – 8:00 p.m.

Participants: Everyone.

Activities performed:

1. Meeting #1.

Activity evidence:

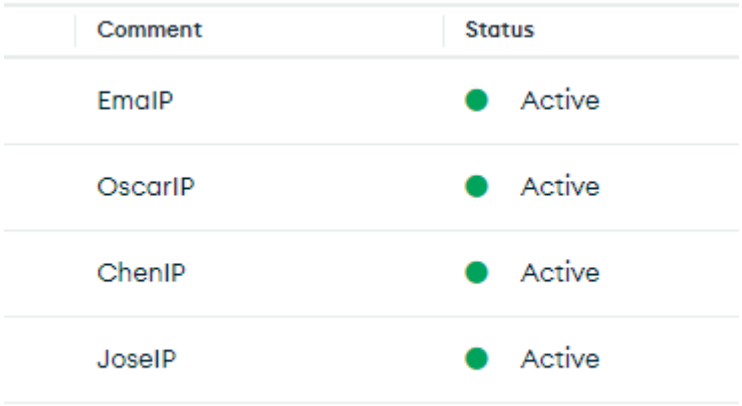


Log entry #2:

Date and time of the activity: October 26, 2023 – 8:00 p.m.
Participants: Sebastian Chen
Activities performed:

1. Created MongoDB database and gave access to all the project members to it.

Activity evidence:

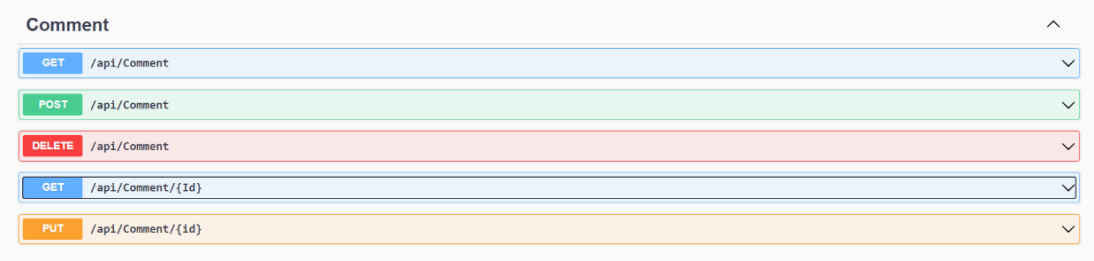


Log entry #3:

Date and time of the activity: October 29, 2023 – 8:00 p.m.
Participants: Sebastian Chen
Activities performed:

1. Created a restAPI to interact with the MongoDB database using .NET C#

Activity evidence:

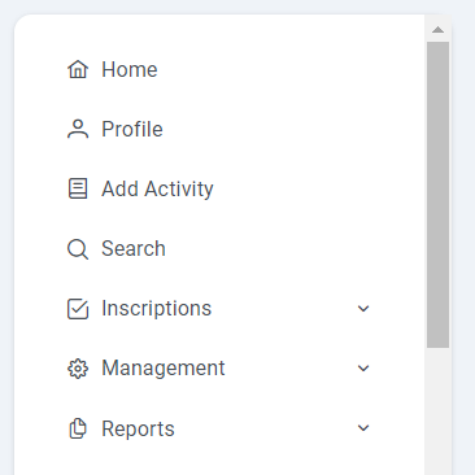


Log entry #4:

Date and time of the activity: October 31, 2023 – 9:00 p.m.
Participants: Emanuel Marín
Activities performed:

1. Added a dashboard and its main views

Activity evidence:

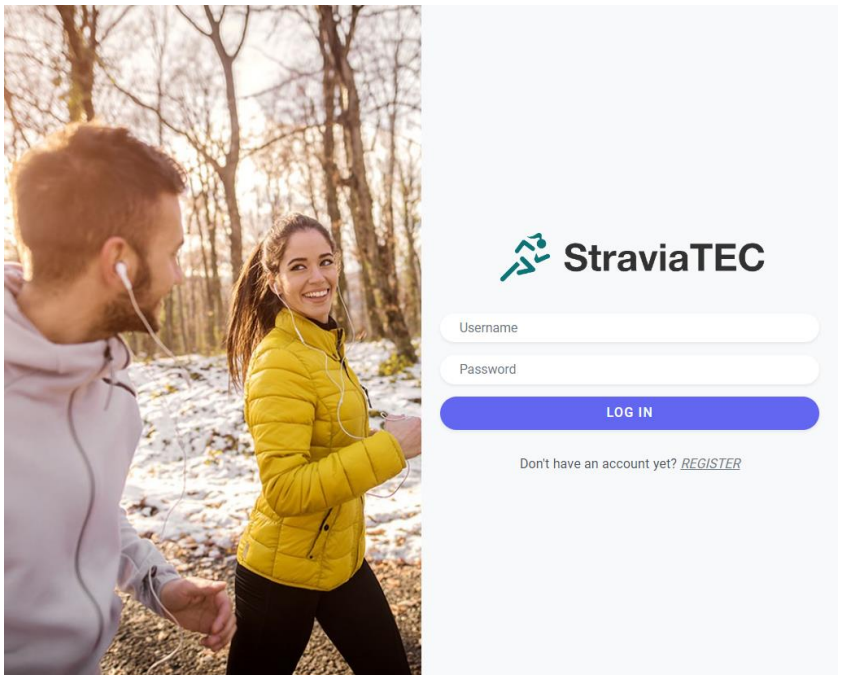


Log entry #5:

Date and time of the activity: November 1, 2023 – 4:00 p.m.
Participants: Emanuel Marín
Activities performed:

1. Added basic login and signup view

Activity evidence:

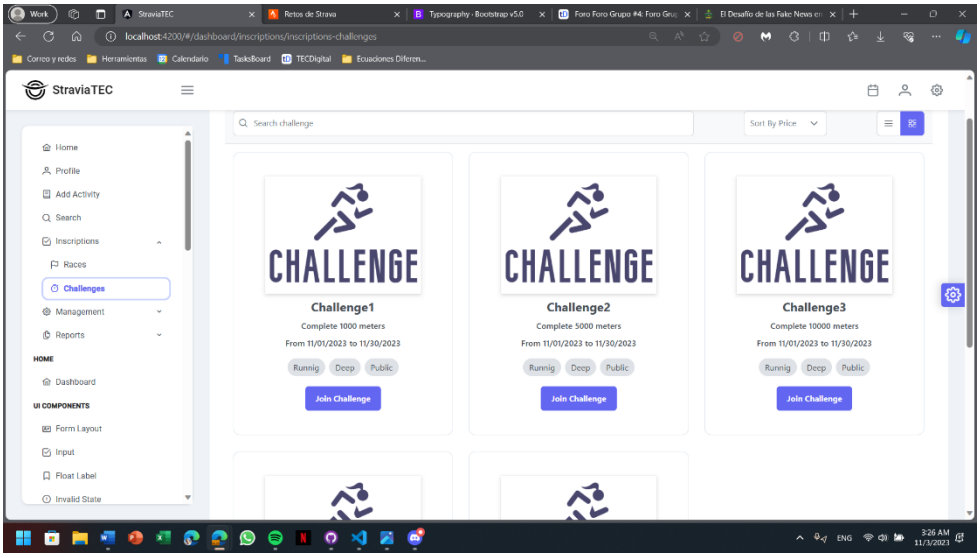


Log entry #6:

Date and time of the activity: November 2, 2023 – 2:00 p.m.
Participants: Andrés Rodríguez
Activities performed:

1. Created an initial view for users and groups searching
2. Created an initial view for challenges inscriptions

Activity evidence:

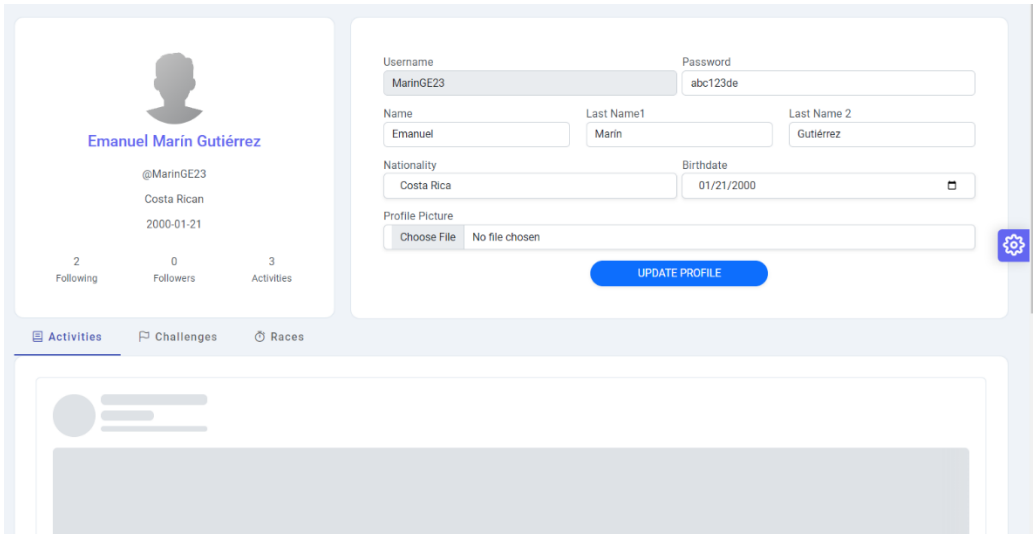


Log entry #7:

Date and time of the activity: November 2, 2023 – 4:00 p.m.
Participants: Emanuel Marín
Activities performed:

1. Create an initial view for profile

Activity evidence:

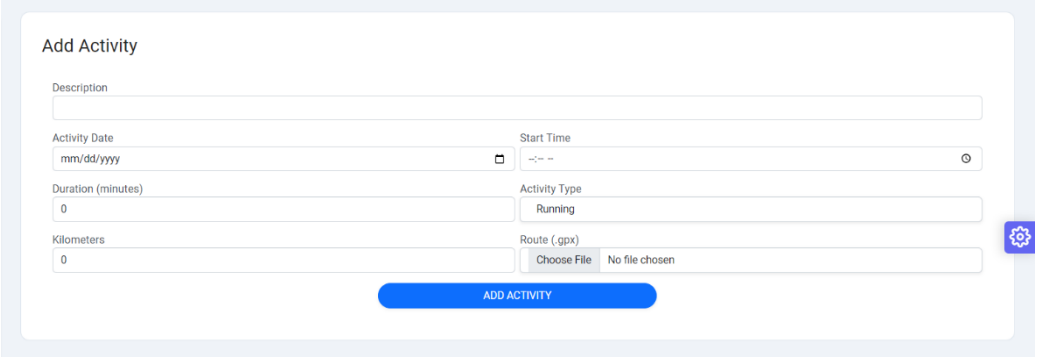


Log entry #8:

Date and time of the activity: November 2, 2023 – 4:00 p.m.
Participants: Emanuel Marín
Activities performed:

1. Create an initial view for add activity

Activity evidence:

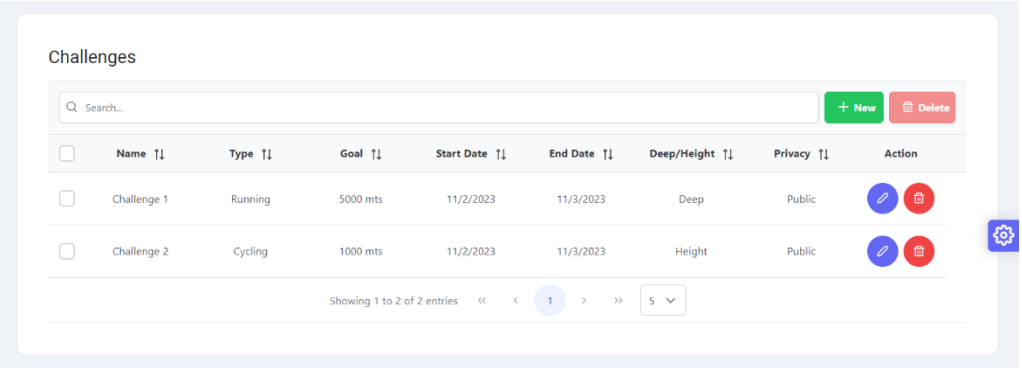


Log entry #9:

Date and time of the activity: November 3, 2023 – 2:00 a.m.
Participants: Emanuel Marín
Activities performed:

1. Created an initial view for challenges management

Activity evidence:

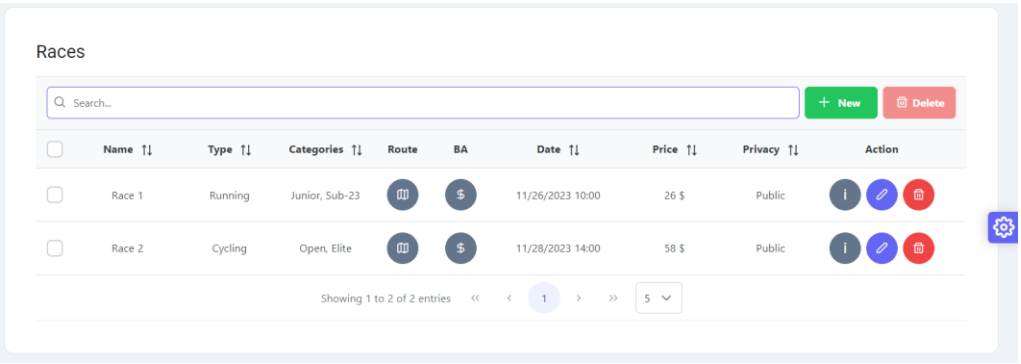


Log entry #10:

Date and time of the activity: November 4, 2023 – 2:00 a.m.
Participants: Emanuel Marín
Activities performed:

1. Created an initial view for races management

Activity evidence:

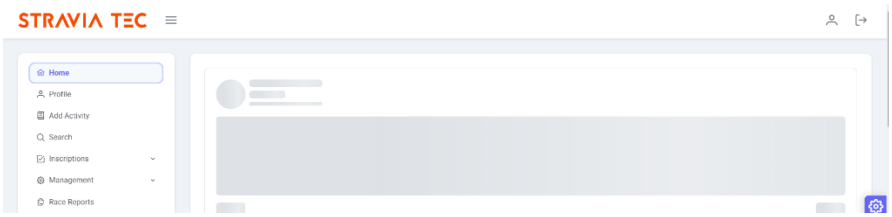


Log entry #11:

Date and time of the activity: November 6, 2023 – 8:30 p.m.
Participants: Emanuel Marín
Activities performed:

1. Created an initial home view with skeleton

Activity evidence:



Activities log

Log entry #12:

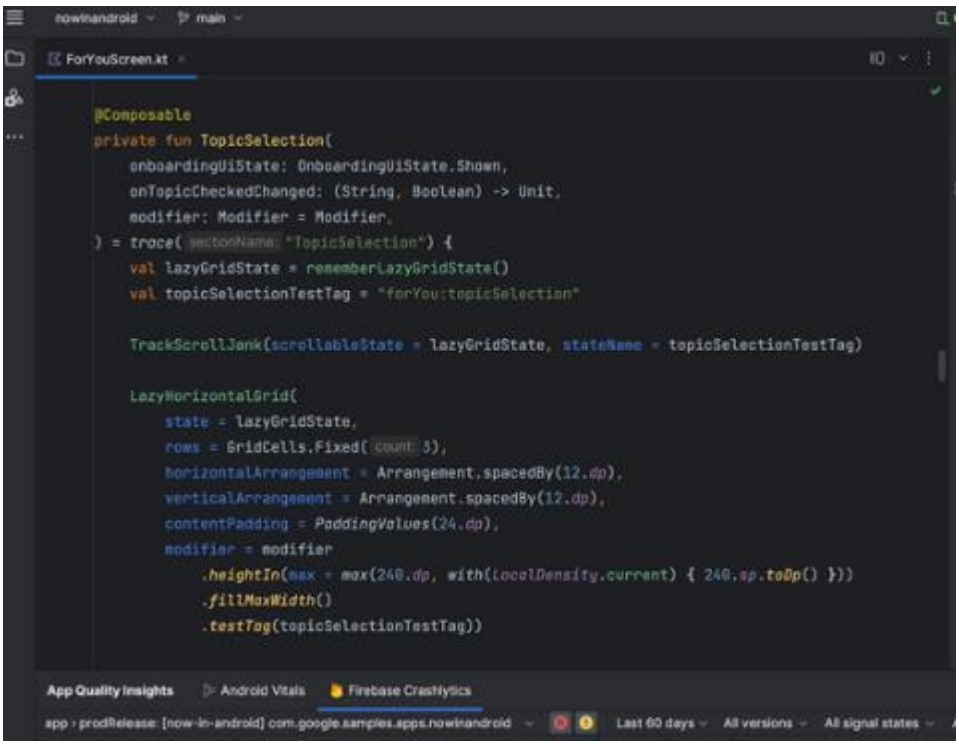
Date and time of the activity: November 8, 2023 – 2:00 p.m.

Participants: Sebatian Chen.

Activities performed:

2. Started the Mobile app creation.

Activity evidence:



Activities log

Log entry #13:

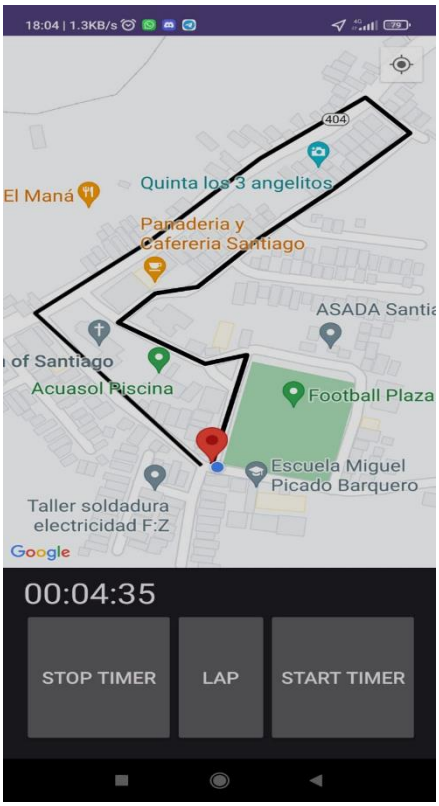
Date and time of the activity: November 9, 2023 – 5:00 p.m.

Participants: Sebatian Chen.

Activities performed:

3. Created views and activities to display a google map and track the user location.

Activity evidence:



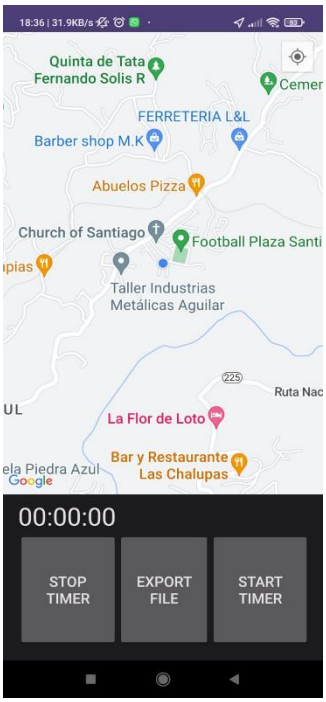
Activities log

Log entry #14:

Date and time of the activity: November 10, 2023 – 8:00 p.m.
Participants: Sebatian Chen.
Activities performed:

4. Creates a gpx file from tracking the user location.

Activity evidence:



Activities log

Log entry #15:

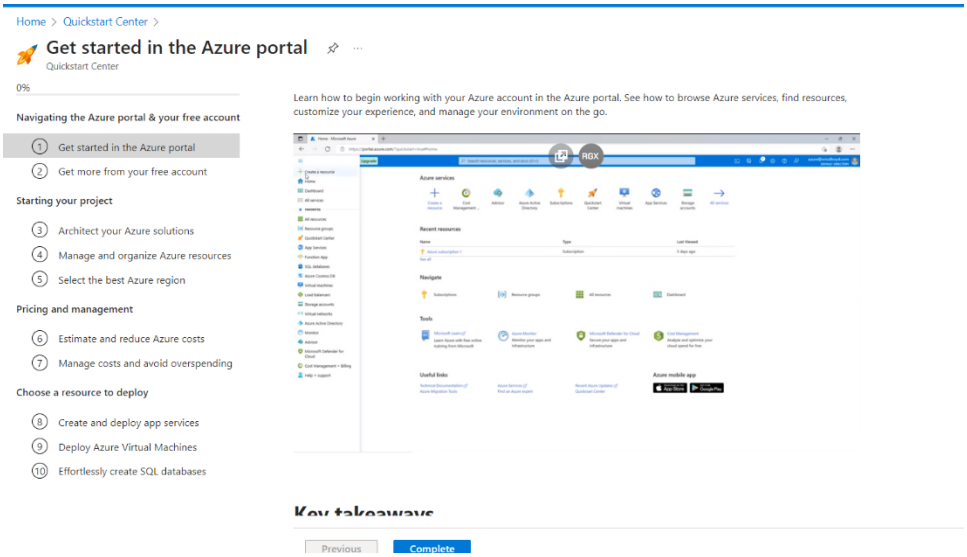
Date and time of the activity: November 10, 2023 – 8:00 p.m.

Participants: Sebatian Chen.

Activities performed:

5. Created and configured a azure portal platform to deploy the project.

Activity evidence:



Log entry #16:

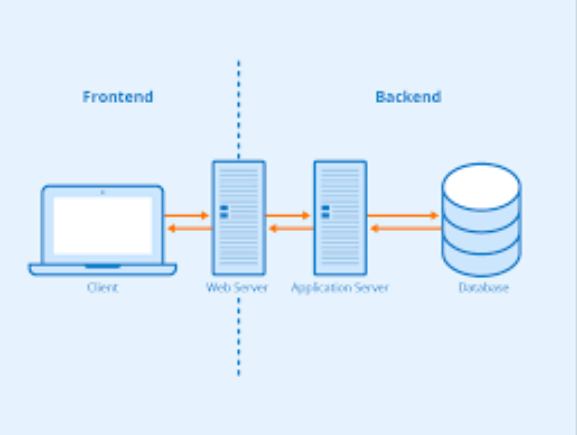
Date and time of the activity: November 11, 2023 – 9:30 p.m.

Participants: Emanuel Marín

Activities performed:

1. I started connecting the previews with the Backend

Activity evidence:



Log entry #17:

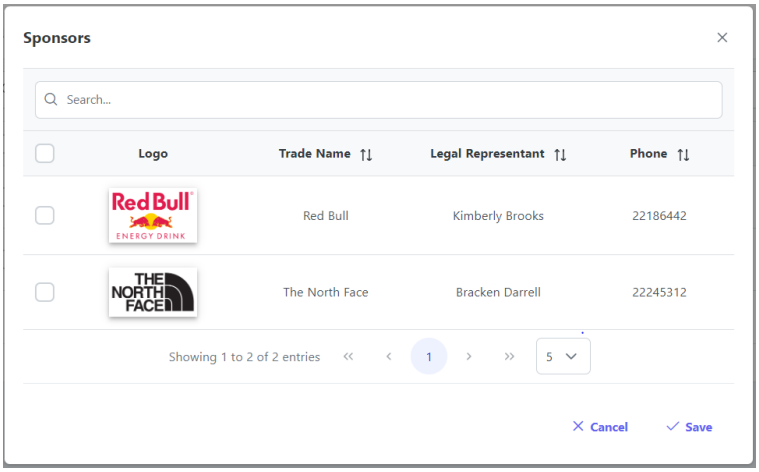
Date and time of the activity: November 15, 2023 – 9:30 p.m.

Participants: Emanuel Marín

Activities performed:

1. I added a dialog window for the sponsors of a race or challenge

Activity evidence:



Log entry #18:

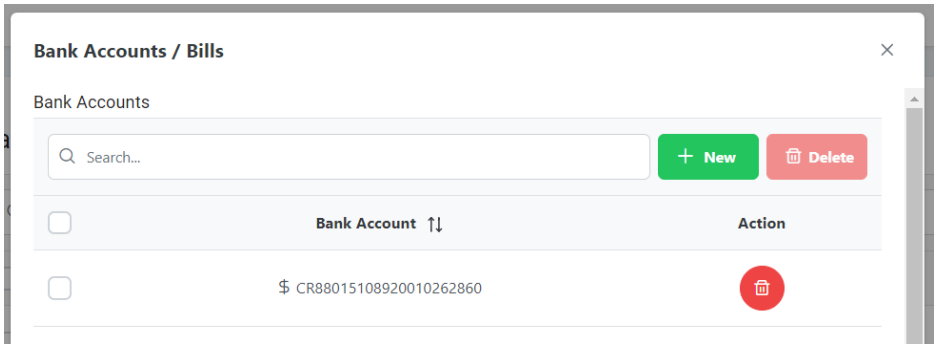
Date and time of the activity: November 16, 2023 – 4:00 p.m.

Participants: Emanuel Marín

Activities performed:

1. Dialog window to add bank accounts for a race

Activity evidence:



Log entry #19:

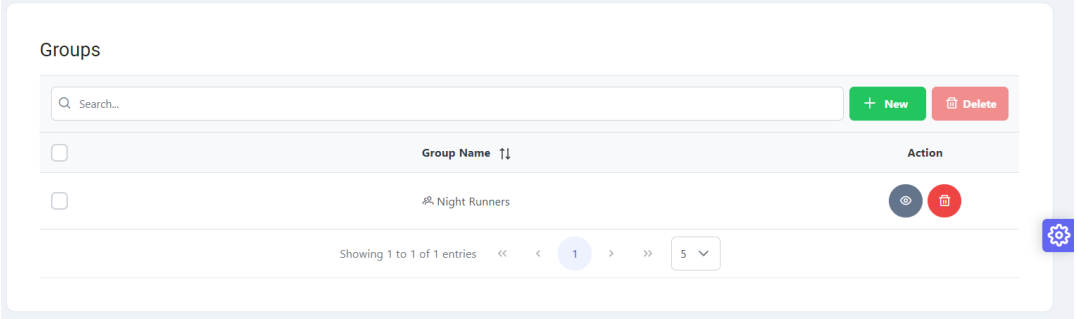
Date and time of the activity: November 16, 2023 – 6:00 p.m.

Participants: Emanuel Marín

Activities performed:

1. Created a view for groups management

Activity evidence:



Log entry #20:

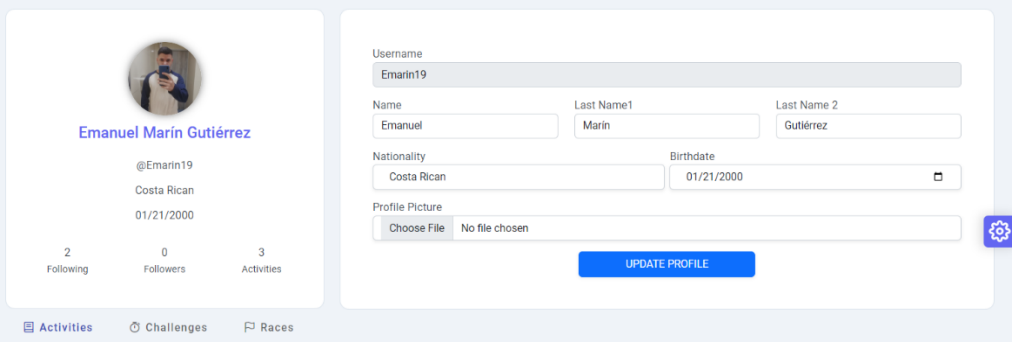
Date and time of the activity: November 17, 2023 – 5:00 p.m.

Participants: Emanuel Marín

Activities performed:

1. I worked on the profile view, connected it to the Backend

Activity evidence:



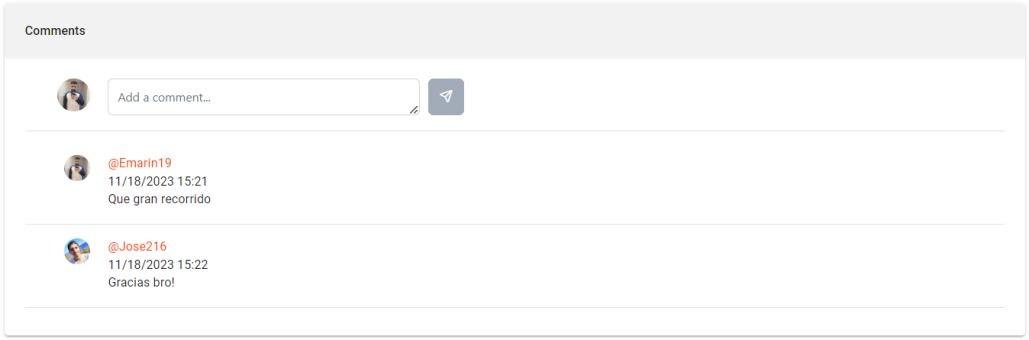
Log entry #21:

Date and time of the activity: November 17, 2023 – 5:00 p.m.

Participants: Emanuel Marín
Activities performed:

1. I worked in the comments section of an activity

Activity evidence:

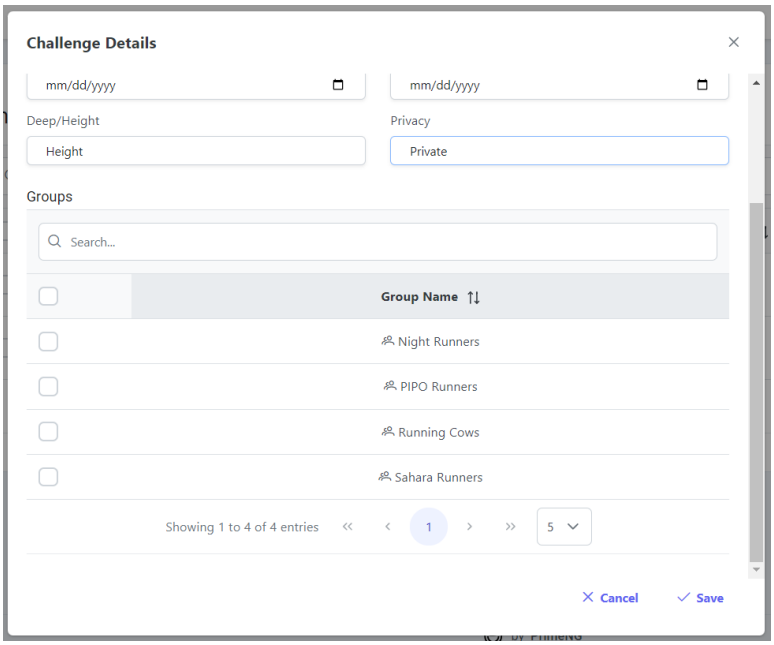


Log entry #22:

Date and time of the activity: November 18, 2023 – 4:00 a.m.
Participants: Emanuel Marín
Activities performed:

1. If the challenge or race is private, you can now select which groups can see it

Activity evidence:

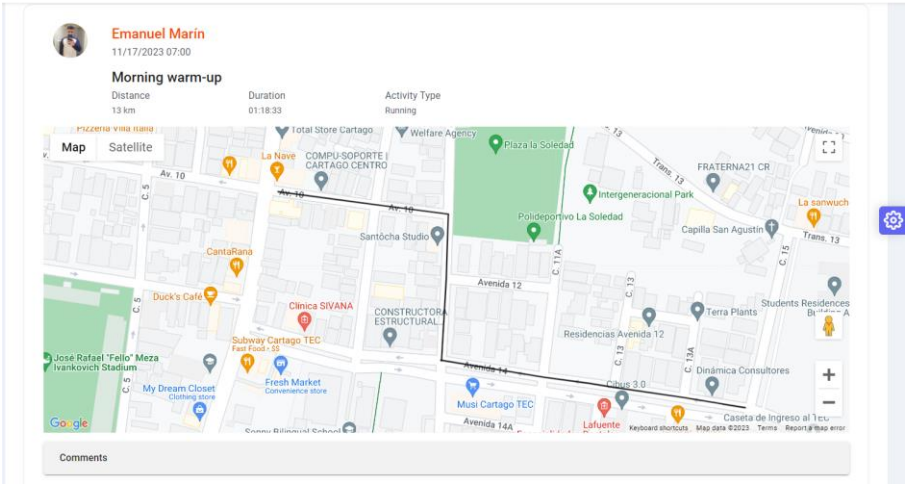


Log entry #23:

Date and time of the activity: November 18, 2023 – 3:00 p.m.
Participants: Emanuel Marín
Activities performed:

1. I finished activity view with map

Activity evidence:



Log entry #24:

Date and time of the activity: November 18, 2023 – 6:00 p.m.

Participants: Emanuel Marín

Activities performed:

1. From race management you can now see the registration request and accept it

Activity evidence:

Log entry #25:

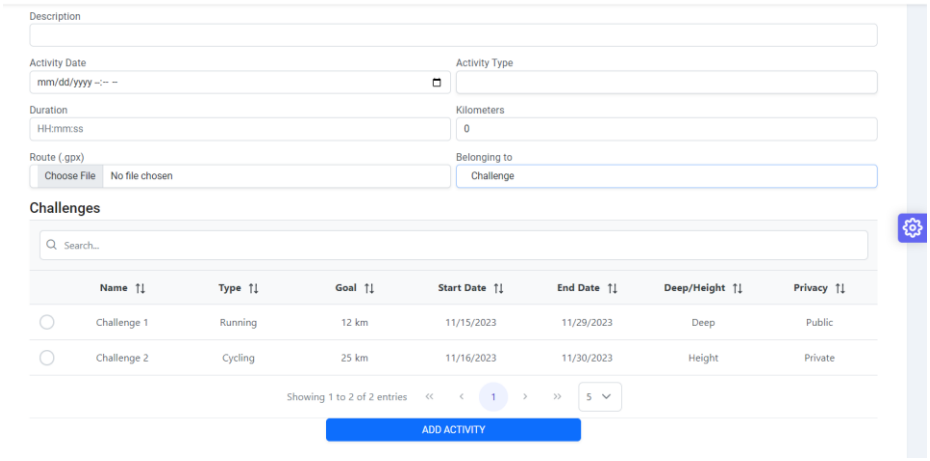
Date and time of the activity: November 19, 2023 – 1:30 p.m.

Participants: Emanuel Marín

Activities performed:

1. When you add an activity, you can indicate whether or not it belongs to a challenge or race

Activity evidence:



Log entry #26:

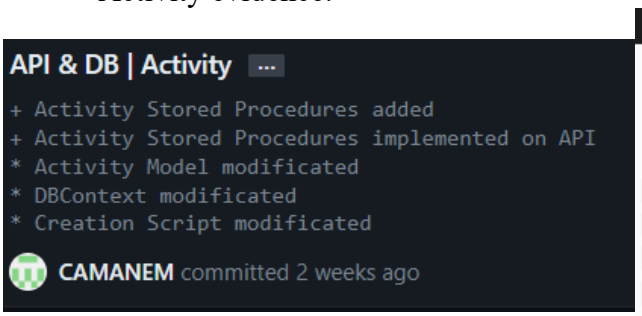
Date and time of the activity: November 03, 2023 – 1:30 a.m.

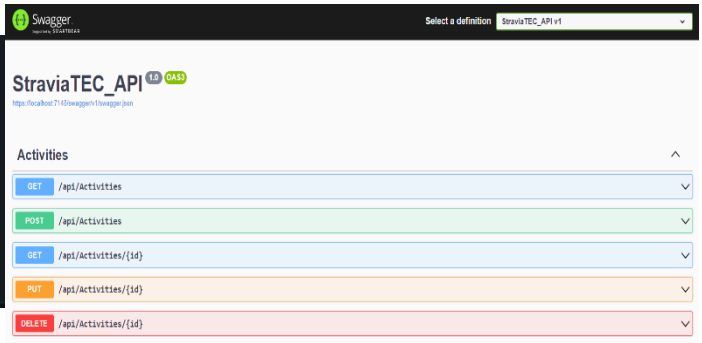
Participants: Óscar Soto

Activities performed:

1. Developed all the Activity CRUD

Activity evidence:





Log entry #27:

Date and time of the activity: November 05, 2023 – 1:30 p.m.

Participants: Oscar Soto

Activities performed:

1. Development of the Groups controller

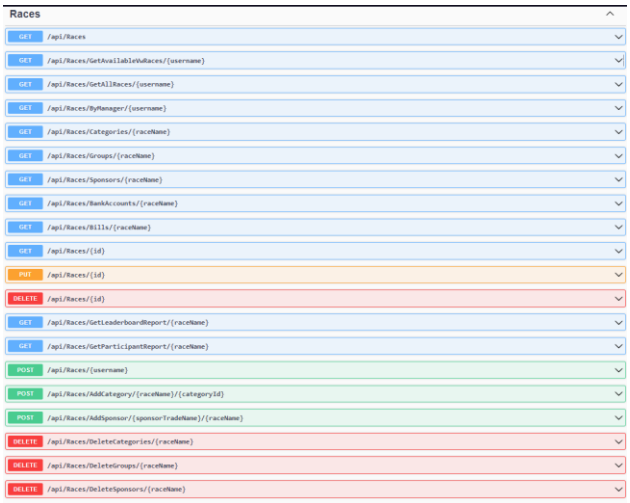
Activity evidence:

Log entry #28:

Date and time of the activity: November 18, 2023 – 2:30 a.m.
Participants: Óscar Soto
Activities performed:

1. Completed the development of the Races controller

Activity evidence:



Conclusions

The integration of PrimeNG dashboards into Angular was very helpful for this project. The smooth collaboration between its components simplified our development, providing a solid and consistent structure.

The strategic fusion of Bootstrap and PrimeNG for design and dashboards offered us a versatile framework. We could build the interface easily and customize our dashboards with distinctive details, because of PrimeNG interface and bootstrap facilities.

Choosing HTML5 as the backbone for our page structure was fundamental for the web development, giving a good user experience and complex views, helped by HTML5's advanced capabilities.

Recommendations

Using the interactive features of PrimeNG to create web dashboards eases the dynamic visualization and management of social content, prioritizing the user experience with interactive elements and real-time updates.

Take advantage of customization options in Bootstrap and PrimeNG to allow users to adapt the interface to their preferences. Bootstrap and PrimeNG provide tools that enable them to customize their experience, from visual themes to widget layout on the dashboards.

Developing an infrastructure, backed by Bootstrap, PrimeNG, and HTML5, is secure and scalable. Implement ordered structures, versatile components and ensure the architecture can handle the future growth of the web-based social network.

Bibliography

Angular. (s. f.). https://angular.io/docs

Capacitor by Ionic - cross-platform apps with web technology. (s. f.). Capacitor. https://capacitorjs.com/

Primefaces. (n.d.). *GitHub - primefaces/sakai-ng: Free Angular Admin Template by PrimeNG*. GitHub. https://github.com/primefaces/sakai-ng

Team, A. C. (s. f.). *Angular material*. Angular Material. https://material.angular.io/