

Project Codebase Documentation & Roadmap

Planning

Proposed idea: Mobile application in which the goal is to conquer as many possible areas on the map by walking. Each walk creates a perimeter of the conquered area and that is the way one wins land. In later development stages it can be extended to multiplayer, and other players may be able to reconquer some areas. We will have a leaderboard with the leading players, based on the conquered area. An eventual integration with AI could be the introduction of an agent for competition in singleplayer mode, one which advises the player about the areas which one should aim for or one which generates areas with bonuses.

I. Proposed Team Structure and Core Responsibilities

The team is split into two specialized units to manage the distinct demands of the mobile application and the backend services.

Role	Name	Focus	Key Responsibility Area
Frontend Developer	Dan Petri	React Native	UI/UX implementation and Frontend Design.
Frontend Developer	Alexandra Moroz	React Native	UI/UX implementation and Database Design.
Frontend Developer	Iris Mihoc	React Native	UI/UX implementation and Frontend Design.
Frontend Developer	Georgiana Asandei	React Native	UI/UX implementation and maintaining project status, goals and communication documents.
Backend Developer	Paul Berindeie	Java	Server setup, CI/CD integration, database migration tools (LiquiBase), project repository creation and setup (backend).
Backend Developer	Ioan Bogdan	Java	API development, project repository creation and setup (frontend).
Backend Developer	Clementina Moisa	Java	API development and LLM Research for future feature integration.

Backend Developer	Larisa Muntean	Java	Server setup, CI/CD integration, database migration tools (LiquiBase).
--------------------------	----------------	------	--

We tried to maintain the original, proposed division of roles, but, because of time constraints, we decided to each focus on certain tasks. This was the final lineup:

Role	Name	Focus	Key Responsibility Area
Frontend Developer	Dan Petri	React Native	UI/UX implementation and frontend design.
Frontend Developer	Iris Mihoc	React Native	UI/UX implementation and frontend design.
Backend Developer / Administrative Tasks	Georgiana Asandei	Java	Mail automation, creating demos and maintaining project status, goals and communication documents.
Backend Developer	Alexandra Moroz	Supabase Java	Database design and management, monitoring service and backend components.
Backend Developer / Backend Architect	Paul Berindeie	Java	Server setup, hosting, CI/CD integration, database migration tools (LiquiBase), project repository creation and setup (backend).
Full Stack Developer	Ioan Bogdan	Java	Development on both backend and frontend components.
Backend Developer	Clementina Moisa	Java	Testing, endpoint implementation and AI feature.
Backend Developer	Larisa Muntean	Java	JWT Authorization for methods, implementing backend components.

II. Proposed Technology Stack

This project uses a modern, performant, and reliable stack, emphasizing consistency, continuous integration, and database reliability.

Frontend Technologies (React Native)

Technology	Purpose	Notes
React Native	Cross-platform mobile application development.	Allows for a single codebase across iOS and Android.
Expo	Simplify mobile development with unified tools.	Uses React Native under the hood. Includes powerful CLI and SDK tools.
API Apple Maps	Location and mapping services.	Essential for core features involving real-world movement and visualization.
Mailchimp	Automated mailing.	Helps automate the process of sending personalised emails to players.

Backend Technologies (Java)

Technology	Purpose	Notes
Springboot	Reduce boilerplate with auto-configuration and starters.	Simplify Java backend development and deployment. Enable rapid, opinionated backend development.
Swagger	Simplify API design and sharing process. Improve developer collaboration on API endpoints.	Includes Swagger UI for visual API testing. Easily integrates with Spring Boot and Node.js.
Java	Core programming language for server-side logic.	Chosen for its robustness, performance, and extensive ecosystem for enterprise applications.
Supabase	Build backend instantly with Postgres and APIs. Accelerate app development with serverless backend tools.	Built on top of PostgreSQL database. Integrates easily with React, Next.js, and Expo.
Render	Deploy web apps and APIs effortlessly in the cloud. Simplify hosting for full-stack applications.	Supports automatic builds on Git push. Includes managed PostgreSQL and background workers.

PostGIS	Add geospatial capabilities to PostgreSQL databases. Enable storage and querying of spatial data.	Extend PostgreSQL for advanced geographic analysis. Ideal for mapping, routing, and geospatial analytics.
LiquiBase	Database Schema Migration Management.	Crucial for versioning the database structure. Ensures that schema changes (tables, columns, indexes) are applied consistently, safely, and reversibly across all development, staging, and production environments.
SonarCube	Static Code Analysis and Quality Gate.	Integrates with Jenkins to continuously inspect code for bugs, security vulnerabilities, and code smells, enforcing high quality standards.
Hugging Face	Large Language Model (LLM).	Used for generating personalised missions for each player.

III. Market Analysis & Competitors

The market is currently dominated by gamified running and habit-tracking apps. Our product must differentiate itself by offering a superior feature set, better user experience or a focused local offering.

Product	Business Model	Key Features / Notes	Competitive Edge & Gaps
Stride	Subscription / Free Challenges	A game in which a player conquers giant hexagons by running (it is not for beginners).	Gap: Previously free features are now paid. Number of players is small, so competitiveness is practically non-existent.
INTVL	Paid / In-App Purchases	High-quality gamified running app with real-world territory capture, global leaderboards, and Strava integration.	Challenge: Directly competes with our core concept. Gap: User feedback indicates the free experience is limited, presenting an opportunity for our free tier. Not available in Europe.

Run An Empire	Game built in Unity	Location-based AR strategy game where running/walking captures hexagonal territory to build an empire.	Challenge: Established product in the gamified niche. Gap: No multiplayer, the other players stealing your areas are fictive. No recent updates.
----------------------	---------------------	--	--

Project development

IV. Available endpoints

1. Zone Operations

Endpoint verb	URL	Description	DTO	Responses
GET	/api/zones/{id}	Get Zone by Id	ZoneDTO	200 – Zone found successfully 404 – Zone not found
PUT	/api/zones/{id}	Update Zone	ZoneDTO	200 – Zone updated successfully 400 – Invalid zone data provided 404 – Zone not found for the given ID
DELETE	/api/zones/{id}	Delete Zone	-	204 – Zone deleted successfully 404 – Zone not found for the given ID
GET	/api/zones	Retrieve all Zones	-	200 – Zones retrieved successfully 500 – Internal server error
POST	/api/zones	Create a new Zone	ZoneDTO	201 – Zone created successfully

				400 – Invalid input data provided
GET	/api/zones/owner/{ownerId}	Get Zones by Owner	-	200 – Zones found successfully for the owner 404 – Owner not found

2. User Operations

Endpoint verb	URL	Description	DTO	Responses
GET	/api/users/{email}	Get User by Email	UserDTO	200 – User found 404 – User not found
PUT	/api/users/{email}	Update User	UserDTO	200 – User updated successfully 404 – User not found
DELETE	/api/users/{email}	Delete User	-	204 – User deleted successfully 404 – User not found
GET	/api/users	Get All Users	-	200 – Users retrieved successfully
GET	/api/users/{email}/stats	Get User Stats by Email	UserStatsDTO	200 – User stats found 404 – User not found

3. Run Operations

Endpoint verb	URL	Description	DTO	Responses
---------------	-----	-------------	-----	-----------

GET	/api/runs/{id}	Get Run by Id	RunDTO	200 – Run found successfully 404 – Run not found
PUT	/api/runs/{id}	Update Run	RunDTO	200 – Run updated successfully 400 – Invalid run data provided 404 – Run not found for the given ID
DELETE	/api/runs/{id}	Delete Run	-	204 – Run deleted successfully 404 – Run not found for the given ID
GET	/api/runs	Retrieve all Runs	-	200 – Runs retrieved successfully 500 – Internal server error
POST	/api/runs	Create a new Run	RunDTO	201 – Run created successfully 400 – Invalid input data provided
GET	/api/runs/username/{username}	Retrieve all runs owned by a specific owner.	-	200 - Runs found successfully for the owner 404 - Owner not found
GET	/api/runs/owner/{ownerId}	Get Runs by Owner	-	200 – Runs found successfully for the owner

				404 – Owner not found
--	--	--	--	-----------------------

4. Auth Operations

Endpoint verb	URL	Description	DTO	Responses
POST	/api/auth/signup	User Signup	AuthDTO	201 – User registered successfully 409 – User already exists
POST	/api/auth/refresh	Refresh Token	AuthDTO	200 – Token refreshed successfully 401 – Invalid refresh token
POST	/api/auth/login	User Login	CredentialsDTO AuthDTO	200 – User logged in successfully 401 – Invalid credentials
PATCH	/api/auth/{email}/change-password	Change Password	-	200 – Password changed successfully 404 – User not found

5. Health-controller

Endpoint verb	URL	Description	DTO	Responses
GET	/api/health/check	-	-	200 – OK
HEAD	/api/health/check	-	-	200 – OK

V. Iteration 1 Goals

This iteration is focused heavily on establishing the foundational infrastructure and design required for all subsequent feature development.

Area	Task/Goal	Owner	Status
Database	Complete the initial relational database design in Supabase.	Ale	DONE
Backend CI/CD, Infra, Setup	Set up the server repository, integrate SonarCube for code quality, and integrate Jenkins for automatic builds. Write initial unit tests and verify the code quality report is generated and visible on the SonarCube dashboard.	Paul, Lari	DONE
Frontend Setup	Setup the frontend repository.	Ioan	DONE
Frontend UX/UI	Finalize the core mobile application design and user flows, in Figma.	Dan, Iris	DONE
Research	Research available Large Language Models and determine the best approach for integrating an LLM (functionality-wise) into our mobile app features.	Clem	DONE
Project Management	Finalize and distribute the initial project documentation.	Georgi	DONE

VI. Iteration 2 Goals

Area	Task/Goal	Owner	Status
Demo	Prepare a script, film and edit the demo.	Georgi, Clem, Ale	DONE
Database	Add polygons (conquered areas) to database.	Ale	DONE
Frontend/Map	Create the map screen, which the user shall interact with.	Lari, Iris, Dan	DONE
Backend	Create Docker container for deployment, DTOs for rest controller.	Paul	DONE
Backend	Create HEAD /api/health for uptime robot	Ioan	DONE
Frontend/Backend Communication	Communication setup between the frontend and the backend.	Ioan, Paul	DONE

Documentation	Document progress on the second iteration.	Georgi	DONE
----------------------	--	--------	------

VII. Iteration 3 Goals

Area	Task/Goal	Owner	Status
Frontend	Fill in zones on map, visual feedback on running (line), warning message for incomplete polygon.	Dan, Iris, Larisa	DONE
Frontend	Authentication by username/email + password: Auth screen + Automatic mail for confirmation	Georgi, Ioan, Dan	DONE
Backend	Authentication by username/email + password: Backend endpoints (JWT tokens)	Paul, Ioan	DONE
Backend	Develop a GET /runs API endpoint that retrieves all runs associated with a specific username. The endpoint should accept a username parameter and return the corresponding run data in JSON format.	Lari	DONE
Backend	Total area sum (accumulated area of all conquered areas).	Ale	DONE
Backend	Write tests for rest controllers.	Paul, Clem	DONE
AI Integration	Research related to AI integration in the application.	Clem	DONE
Documentation	Document progress on the third iteration.	Georgi	DONE

AI Integration research:

Ideas for challenges:

1. Perimeter/area of the zone > a given value
2. Conquer any zone in x minutes or less
3. Conquer a zone that touches/intersects at least x zones you already own
4. Conquer a zone that is at least x meters away from all your zones
5. Conquer a zone in an x shape

Ideas for bonuses:

1. The zone is locked for x hours after being conquered – no one else can conquer it for x hours
2. The zone has double, triple, etc. points
3. The zone is harder to conquer (the opponent must conquer it twice)

AI integration consists of API calls to HuggingFace to the Llama-3B-Instruct LLM.

It will be trained to generate missions and bonuses in JSON format, based on the provided context.

The context includes the following aspects:

- User level (optional)
- User play style (chosen at sign-up): Strategist, Conqueror, Relaxed, Runner, etc.
- User's recent history (strings such as): conquered_x_zones, lost_x_zones
- Conditions of the walking/running session: time, weather (optional), current location type (park, city center, industrial area, countryside, mountains, etc.)

The LLM will generate a JSON of the form:

```
{  
  "mission_name": "Creative Name",  
  "flavor_text": "Short catchy passive-aggressive encouragement words.",  
  "challenge": {  
    "type": "ONE_OF_COND_TYPES",  
    "target_value": 100  
  },  
  "reward": {  
    "type": "ONE_OF_REW_TYPES",  
    "target_value": 2.0  
  }  
}
```

This JSON will be consumed by the backend, and after a zone is conquered, the system checks whether the mission has also been completed and updates the corresponding context.

Explanation + Example: The idea is to have an LLM that generates missions and bonuses when a mission is completed. When the user presses Start Run, they receive 3 missions and can choose one.

Here is a test JSON generated in Java, which works satisfactorily for when a player starts running in the city center:

```

{
  "mission_name" : "Urban Oasis",
  "flavor_text" : "Create a haven amidst the concrete jungle.",
  "challenge" : {
    "type" : "COND_TIME",
    "target_value" : 60
  },
  "reward" : {
    "type" : "REW_LOCK",
    "target_value" : 2.0
  }
}

```

- COND_TIME = conquer any zone in less than 60 (target value) minutes
- REW_LOCK = the zone is immune to attacks for 2 hours

VIII. Iteration 4 Goals

Area	Task/Goal	Owner	Status
Demo	Prepare a script, film and edit the demo.	Georgi, Clem, Ale, Lari, Iris	DONE
Backend	Clean up the backend.	Ale	DONE
Backend	Protect routes with JWT.	Lari	DONE
Backend	Endpoints- top players (leaderboard), user total area	Clem, Ale	DONE, DONE
Backend	Moving the logic from controllers to services.	Ioan	DONE
Backend	Authentication (login, auth, refresh token); refactoring users' server	Paul	DONE
Frontend	Overlay total area + new run area	Iris, Dan	DONE

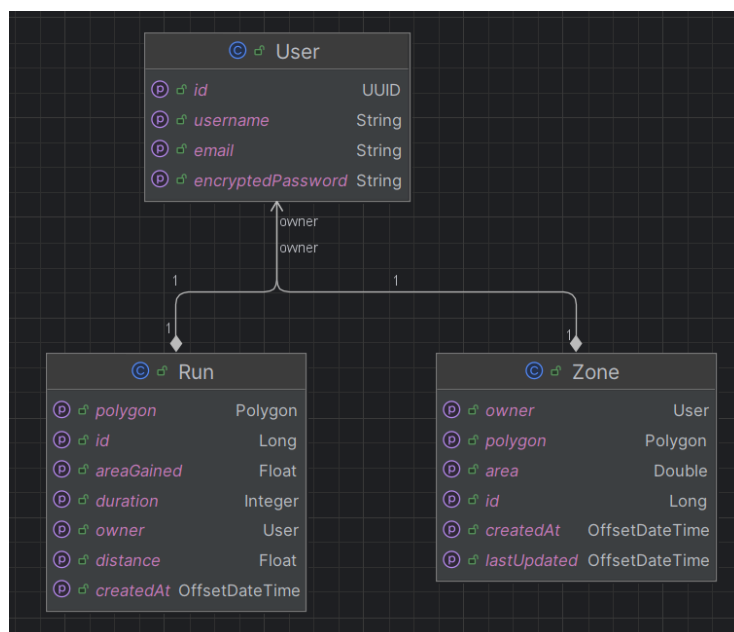
Documentation	Document progress on the 4th iteration.	Georgi	DONE
---------------	---	--------	------

IX. Iteration 5 goals

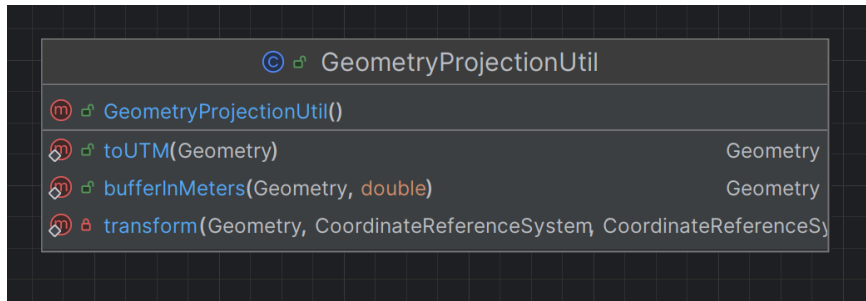
Area	Task/Goal	Owner	Status
Frontend	Integration with Websockets, leaderboard update.	Dan, Iris, Ioan	DONE
Backend	Websockets Implementation, new run websocket.	Paul, Lari	DONE
Backend	Automated emails containing personalised missions for players.	Georgi, Clem	DONE
Presentation	Research for the project's pitch in front of the jury.	Georgi	DONE
Documentation	Document progress on the 5th iteration. Added diagrams to documentation.	Georgi, Ale	DONE

X. Class Diagrams

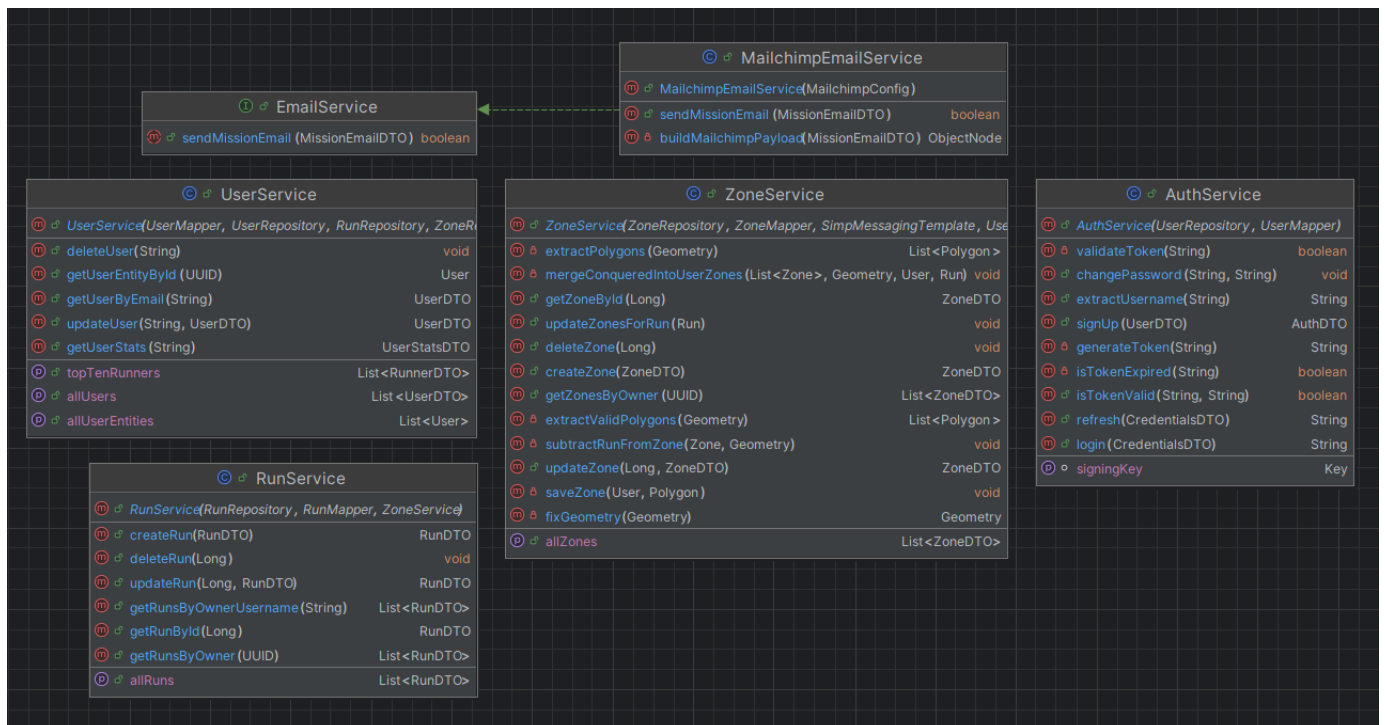
Backend – Domain Class Diagram



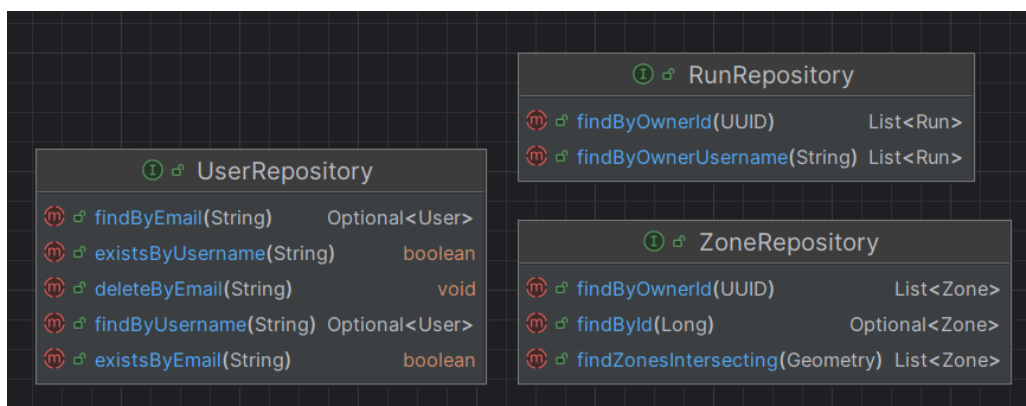
Backend – Utils Class Diagram



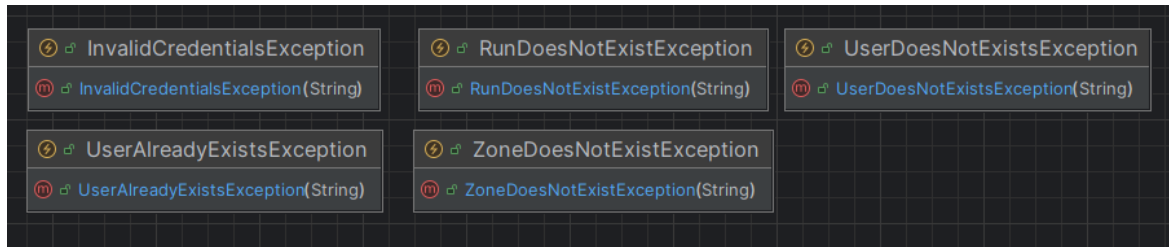
Backend – Service Class Diagram



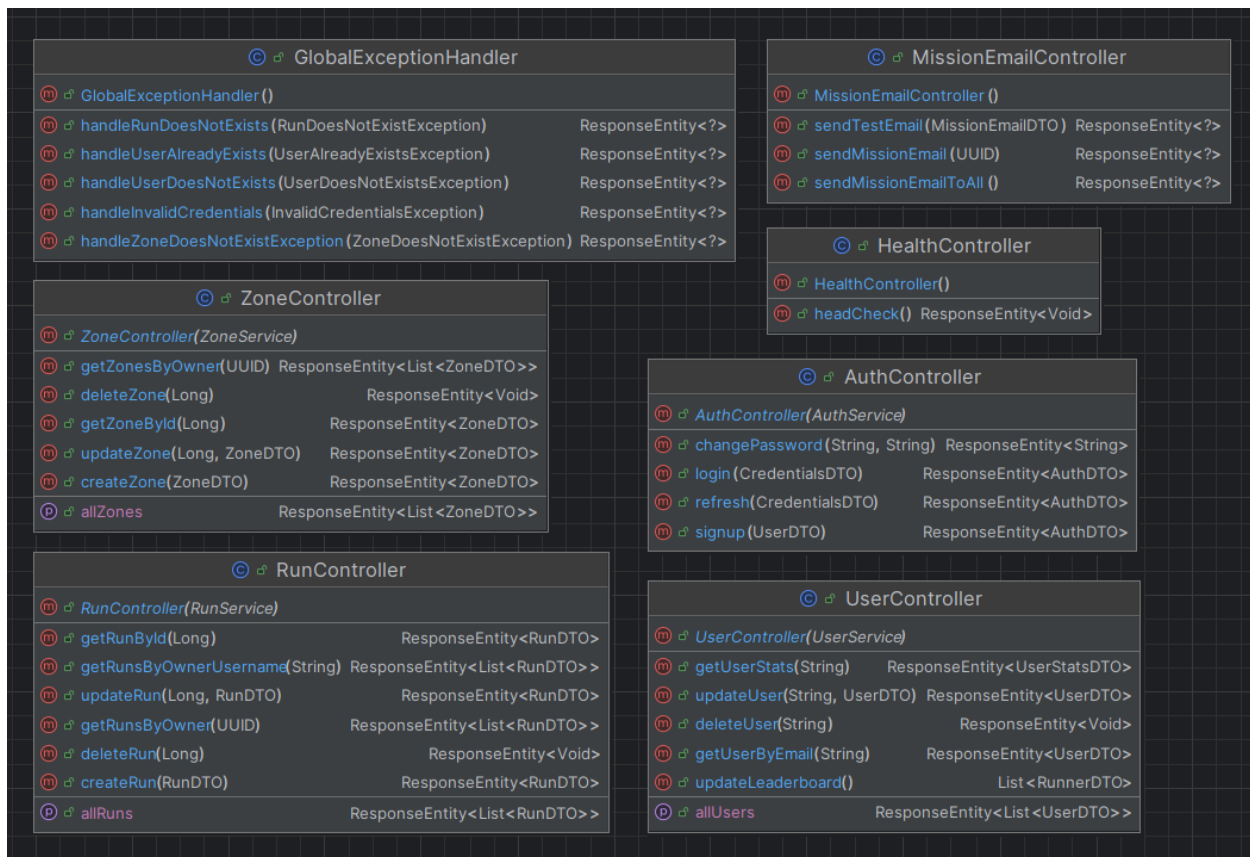
Backend – Repository Class Diagram



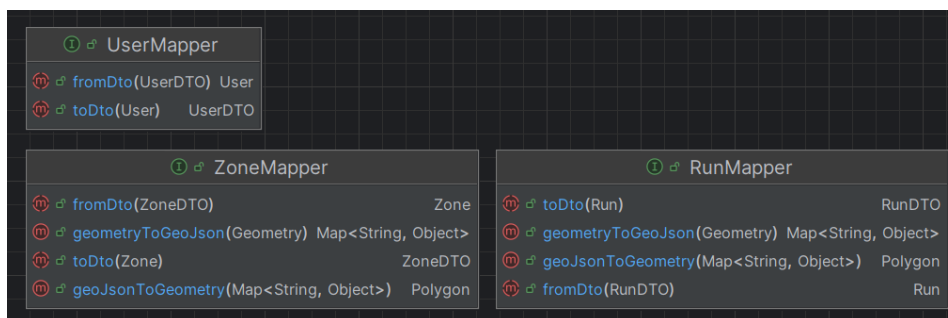
Backend – Exceptions Class Diagram



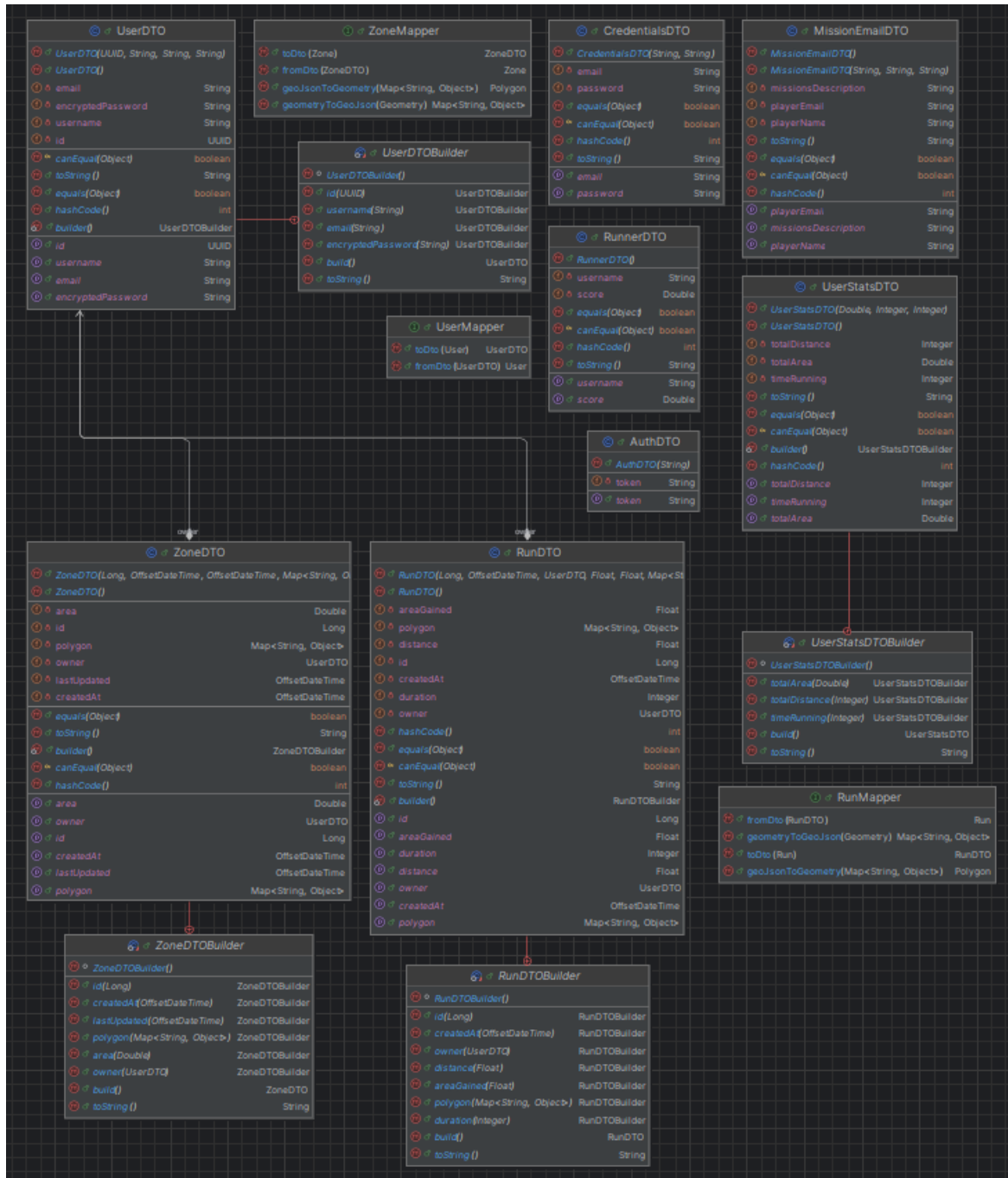
Backend – Controllers Class Diagram



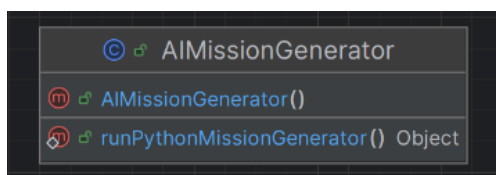
Backend – Mappers Class Diagram



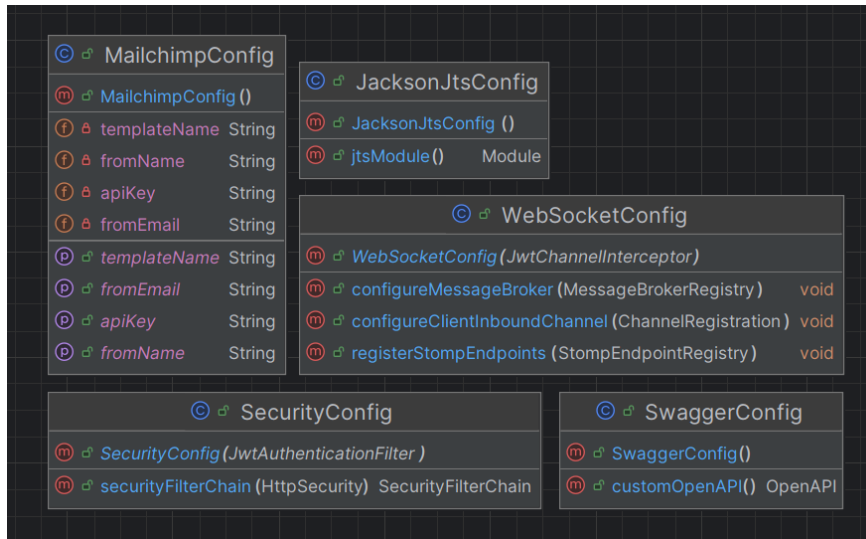
Backend – DTOs Class Diagram



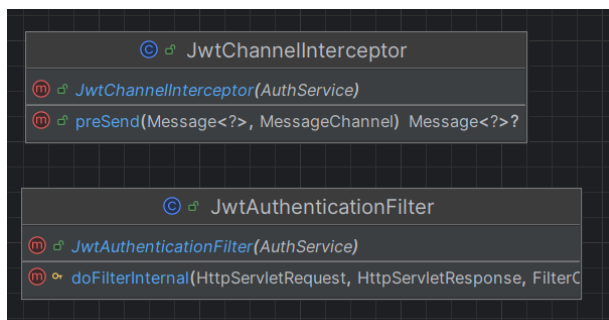
Backend – AI Class Diagram



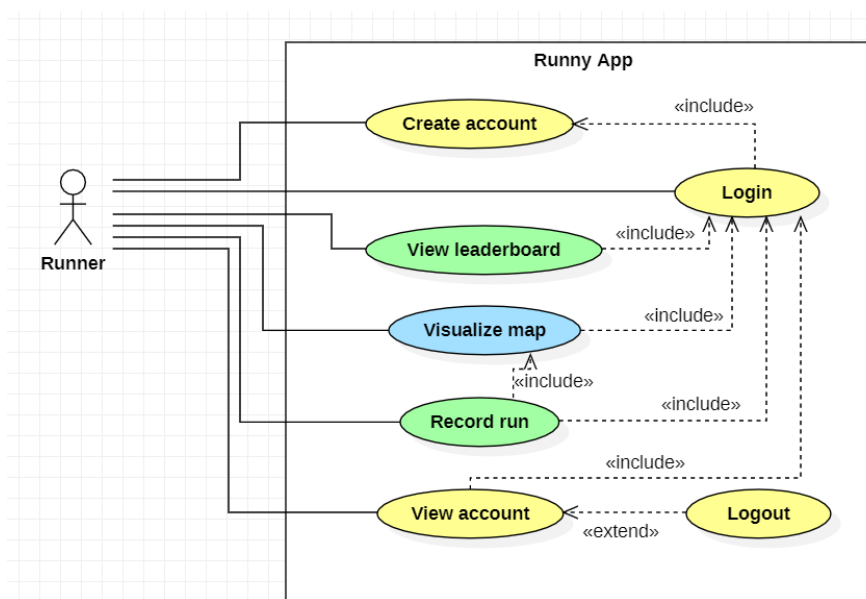
Backend – Config Class Diagram



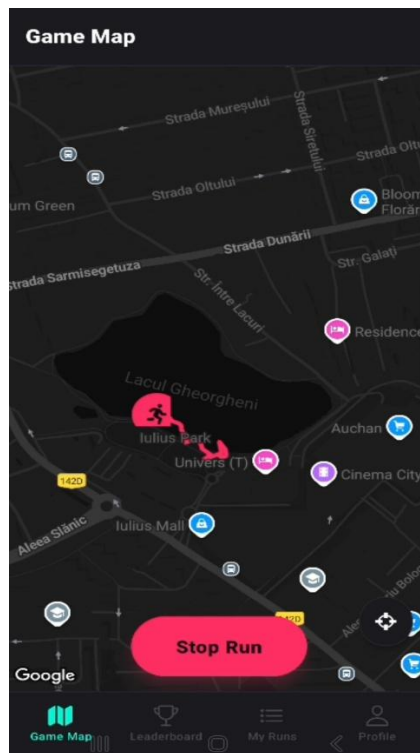
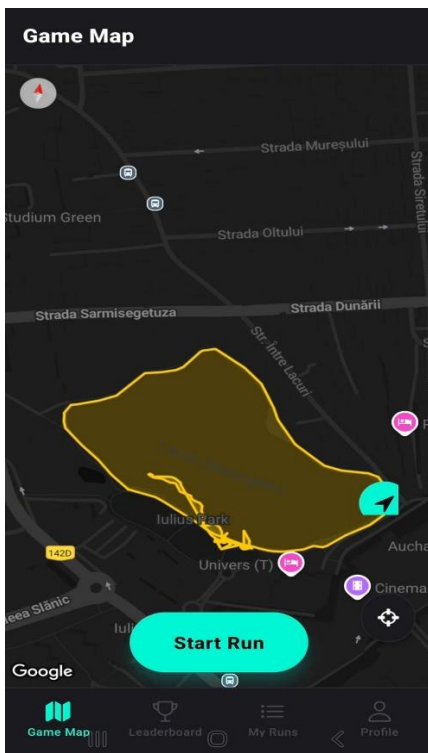
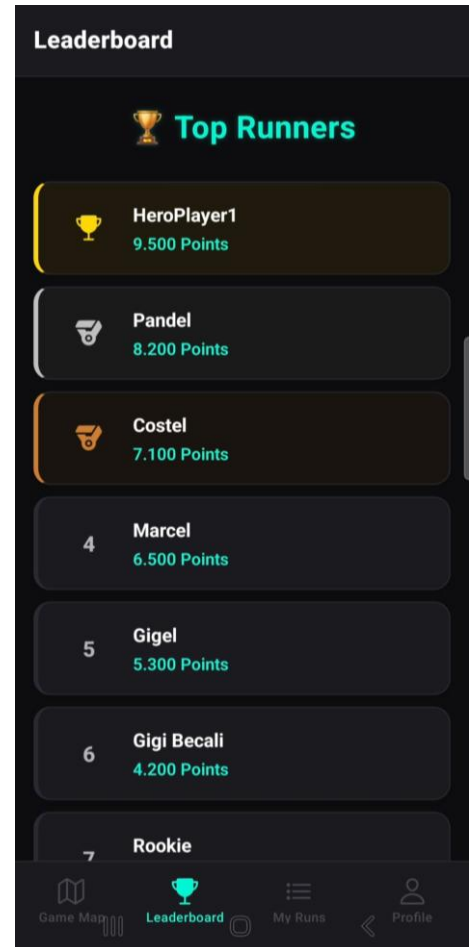
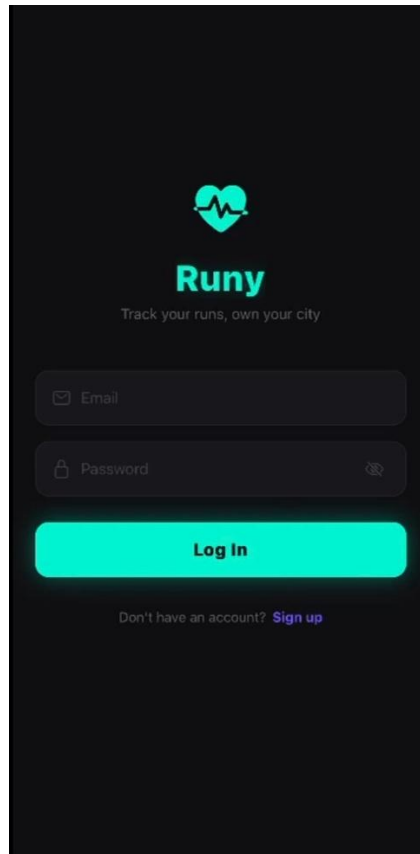
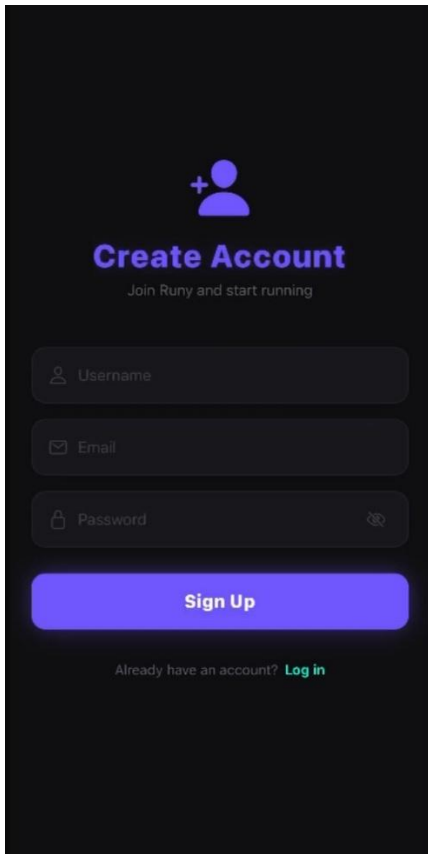
Backend – Security Class Diagram



XI. Use Case Diagram



XII. Frontend screenshots – Sign up, Log in, Conquered Area, Live Tracking, Leaderboard



XIII. Team journey – meetings

1. Online meeting on 3.10.2025 – topics:

- introductory meeting with the mentor of team G05

2. Online meeting on 13.10.2025 – topics:

- finalisation of the first tasks on the CodeOfTalent platform (team identity)

3. Physical meeting on 18.10.2025 – topics:

- division on frontend, backend and tasks by the team members
- technologies discussed: Azure hosting for server (<https://azure.microsoft.com/en-us/pricing/offers/ms-azr-0170p>), Postgress, NeonTech for DB hosting (neon.tech)
- ideas for the project: power-ups in specific points on the map, obstacles, multiplayer

4. Online meeting on 29.10.2025 – topics:

- finalisation of the project idea description task on CodeOfTalent
- discussions relating to the progress on each team member's tasks
- presentation of ideas relating to integration of AI
- presentation of authentication functionalities provided by Supabase
- presentation of documentation progress

5. Online meeting on 31.10.2025 – topics:

- updated the team lead on each member's progress
- we should start with bonuses (Ai-wise)
- for documentation: screenshots, things that highlight the application
- functional map for next iteration, leaderboard for later but something like a screen would be good for the next one
- division of tasks for next iteration

6. Online meeting on 09.11.2025 – topics:

- discussed ways to handle the hosting for backend: Render vs. Railway vs. Azure, team settled on Render

- discussed ways to approach the demo, scripts and functionalities to highlight in the video: in the demo we will highlight the user running, the conquered areas and the movement of the user with a moving dot
- decided on holding more frequent meetings up until the finalisation of the demo
- updates on progress, team has viewed the frontend and backend-wise there will be new endpoints added based on necessity

7. Online meeting on 10.11.2025 – topics:

- to implement: GET /runs based on the username (perhaps Unique username in database); page for user choice; HEAD /api/health for uptime robot
- came up with various ideas for the demo, set a date and time for filming

8. Online meeting on 12.11.2025 – topics:

- edited and finished the first demo using Microsoft's Clipchamp software

9. Online meeting on 14.11.2025 – topics:

- Retro board – discussed likes/dislikes in terms of project development; Retro board to improve: communication, more frequent meetings
- for next sprint: auth + run saving + area calculation
- AI integration research for generating bonuses
- JWT for authentication
- warning message for incomplete polygons + other validation
- presentation of up-to-date documentation

10. Online meeting on 28.11.2025 – topics:

- updated mentor on each team member's progress
- set tasks for next iteration
- Demo expectations: Auth, Multiplayer, Frontend polishing, JWT protection for endpoints

11. Online meeting on 28.11.2025 – topics:

- settled on functionalities to be promote in the second demo
- recapped the things discussed at the previous meeting

12. Online meeting on 8.12.2025 – topics:

- filmed, edited and finished the demo using Microsoft's Clipchamp software

13. Online meeting on 12.12.2025 – topics:

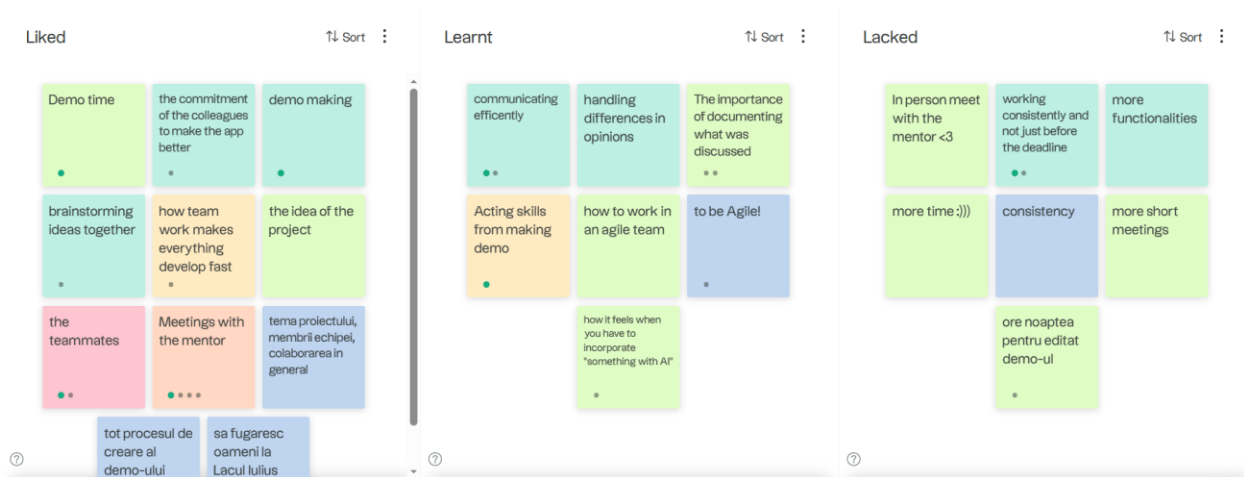
- discussed various technical topics:
 - bonuses shall be implemented in the future (the leaderboard is based on total conquered area and not on points, so this would require to modify the backend)
 - alternative for above: we could use a blocking system for the conquered land
 - integrating AI for mails
 - visual improvements on frontend, web sockets, notifications

14. Online meeting on 11.01.2026 – topics:

- future implementations will concern notifications when player's area gets conquered
- discussed automation of sending emails with custom missions for each player
- planned a physical meeting with all team members to discuss the project presentation

15. Online meeting on 14.01.2026, final meeting with the mentor – topics:

- brought the mentor up to date with all the additions to the project and discussed facts on the 11th of January.
- feedback with Retrotool + farewell before the final Project Presentation



16. Physical meeting on 15.01.2026 – topics:

- Planned presentation layout and chose presenters

XIV. Future improvements

- Integration with AI:

- the introduction of an agent for competition in singleplayer mode, one which advises the player about the areas which one should aim for or one which generates areas with bonuses;
 - automatic generation of places with bonuses, based on the high traffic in those specific areas;
- **Notifications:** for players which have lost previously-conquered areas;
- **Manually adding areas:** possibility to add specific areas in the app with bonus points, thanks to them concerning a touristic/historical attraction, or an area of interest.