

Project Plan

Details

Date: 27/10/2024

Course: BLG411E - Software Engineering

Group Name: Witch

Project Name: Puzzles

People & Roles:

- Ertuğrul Şentürk - Backend Development - Project Leader
- Utku Biçer - Frontend Development - Design
- Hüseyin Şimşek - Backend Development - Puzzle Algorithm Developer
- Oğuzhan Altıntaş - DevOps & Backend Engineering
- Oğuz Eren Kacar - Backend Development
- Lucas Holczinger - Frontend Development - Design
- Ata Türköz - Frontend Development

Project description:

Our main goal is to create a platform to promote logic puzzles and games that are less-known or created by us, to support this community of puzzle enthusiasts.

We will be creating a repertoire of logic puzzles, where we are creating the algorithms for puzzle creation, solving and categorizing.

For this project, we are only going to be extending the repertoire for Sudoku. We will create a solver, a generator, and a categorizer for Sudoku puzzles.

After we have the repertoire ready, we will create a backend server and populate the puzzles and register them to our database. We will create a RestAPI and deploy it on Convex.

Then, we will create a simplistic website to accommodate these puzzles and our leaderboard for the registered users to showcase their time of solving.

Tech Stack:

- Design: Figma
- Frontend: Svelte Kit, TypeScript
- Backend: FastAPI, SQLAlchemy, Python3.12
- Database: Alembic, PostgreSQL
- External Services: Firebase for Authentication
- Deployment: Vercel (Frontend), Convex (Backend) (Not decided yet)
- CI/CD: GitHub Actions
- Tests: PyTest (Backend)
- Communication and Project Management: GitHub

- Documentation: In GitHub, using Markdown & Swagger
- API: GitHub API

Technical Deliverables

Labels:

- **(no code)** label means that the task is not directly related to coding, but rather planning, designing, or discussing the code
- **(abstract code)** label means that the task is related to coding, but rather abstract and not directly related to the implementation of the project, like creating the structure of the project with all the necessary files, methods but leaving them blank
- **(X)** X being an number, means X man.weeks of work
- **(SIZE)** SIZE being the size value, means the size of the task, could be XS, S, M, L, XL

Design

General Design (4)

Note: Everyone should be able to see the design and approve it

- Deciding color palette and themes for the project (S)
- Creating the general user design and user experience in Figma for landing page (L)
- User experience of the Sudoku page, this design should consist of all the animations/transitions that will be used for Sudoku's solving user experience ``XL``
- Designing the leaderboard page (L)

Backend

Puzzle Libs (4)

- Creating a general environment for Sudoku with interfaces and visualizations (M)
- Creating the solver for Sudoku (M)
- Creating the generator for Sudoku (XL)
- Creating the categorizer for Sudoku, to determine the difficulty (L)
- Isolating the libs for the use in FastAPI and transferring them to the backend repository (XS)

FastAPI (6)

- Creating the endpoint documents that will be used, deciding the endpoints, request and response type **(no code)** (M)
- Database schema **(no code)** (M)
- Creating the general structure of the backend (services, models, routers etc) **(abstract code)** (M)
- Creating base unit tests (XL)
- SQLAlchemy entities and crud operations (XL)
- Alembic for database migrations (L)
- Authentication services for Firebase (L)
- Implementing FastAPI endpoints (L)
- Creating Swagger documentation (M)

Frontend

Authentication (2)

- Authenticate users with Firebase Auth (M)
- Research JWT-based auth with Firebase (if necessary) (S)
- Store user data for leaderboard system (L)

Landing Page (2)

- Implementing the design from Figma to Svelte (L)
- Login/registration options (using Firebase) (L)
- If already authenticated redirect to puzzle page (XS)

Puzzle & Leaderboard Page (4)

- Implementing the design from Figma to Svelte (XL)
 - Option to select difficulty (easy, intermediate, hard)
 - Fetch random puzzle and display its unique id
- Easy to use interface / UX when solving the puzzle (L)
 - For Sudoku accept number key inputs (single digit)
- Implementing the design for our leaderboard, displaying it below/beside the puzzle or with hidden behind a toggle button (XL)
- Validate if the puzzle is solved on the client-side (M)
- Get the repository star count information from GitHub API and display it (M)

Deployment & DevOps

Backend (3)

- Deployment of the backend on Convex (XL)
- Adding unit tests to automatically run everytime a new push comes (L)
- Continues Deployment pipeline in GitHub Actions for the backend (M)

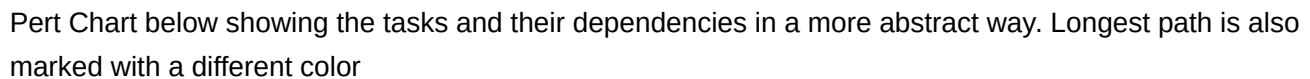
Frontend (3)

- Deployment of the frontend on Vercel (L)
- Adding CI/CD pipeline in GitHub Actions (L)

Charts & Timeline

All the charts here are created by the Mermaid library, you can see the code for the charts [here](#).

Gantt Chart below is the graphical representation of the project plan. It shows the tasks, their start and end dates, and the dependencies between them.



- **Lack of communication:** Since we are working remotely, we might face some communication issues. We will have to make sure that we understand each other and communicate effectively.
- **Lack of user feedback:** For the best user experience for a puzzle platform, we need user feedback. During the time of main development, we might not have enough users to give feedback.
- **Lack of funding:** Since this is an open-source project, we might not have enough funding to extend and host the project for a broader audience.