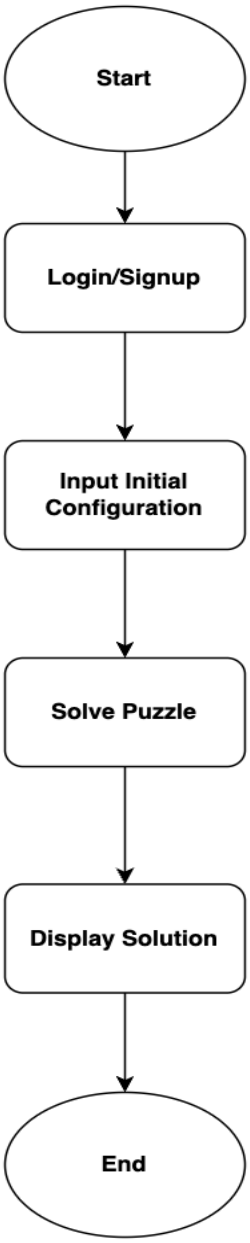
Project Design Specification Individual Flask Project

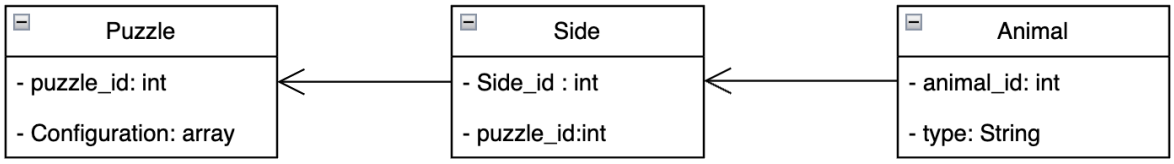
Project Description:

The end product is a puzzle-solving application for the wolves and sheeps problem, focusing solely on the movement of wolves and sheeps across a river using a boat. The application will provide users with an interactive platform to input initial configurations, solve the puzzle, and view solutions.

Flow Chart:



ERD Diagram:



Methods:

The wolves and sheeps puzzle solver utilize brute force, heuristics, backtracking, and constraint satisfaction methods. Brute force explores all possibilities, heuristics prioritize moves, and backtracking systematically explores configurations. Constraint satisfaction models moves iteratively, while optimization techniques like memorization enhance solver efficiency. These methods collectively enable efficient puzzle-solving strategies.

Market Space and Selling Points:

The application caters to puzzle enthusiasts, logic game players, and educators interested in problem-solving and critical thinking. Its selling points include:

Engaging and challenging puzzle-solving experience.
Educational value in promoting logical reasoning and strategy development.
User-friendly interface for easy navigation and interaction.

Features:

1. User-friendly interface for inputting puzzle configurations and viewing solutions.
2. Validation of input to ensure the puzzle is solvable and conforms to game rules.
3. Display of step-by-step solution of puzzle-solving process.
4. Option to save and load puzzle configurations for future solving.

Bonus: Multiplayer Features (10% bonus points):

1. Real-time multiplayer functionality allowing users to collaborate or compete on solving puzzles.
2. Chat feature for communication between players.
3. Leaderboards to track high scores and achievements.

Deployment:

Major steps to deploy a Flask application on Heroku:

1. Prepare Your Flask Project:

   Ensure your Flask project is structured correctly with necessary files like app.py, requirements.txt, and any static or template files.

2. Create a Heroku App:

   Log in to your Heroku account and create a new app using the Heroku dashboard or the Heroku CLI.

3. Set Up Git and Initialize a Repository:

   Initialize a Git repository in your project directory and commit your files to the Git repository.

4. Deploy Your Flask App:

   Connect your local Git repository to your Heroku app and push your code to Heroku using Git: git push heroku main.

5. Configure and Scale Your App:

   Set any necessary environment variables using the Heroku dashboard or CLI.

Milestones:

M1: Basic Flask application setup with input interface and constraint validation.
M2: Implement the puzzle solving algorithm and display solution steps.
M3: Error handling, save/load functionality, and hint system.
M4: Testing, debugging, and optimization for performance.
M5: Deployment & Bonus multiplayer features implementation.