UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and Information Technology

System Requirements Specification
Cerebero

Frederick Ehlers	11061112
Jacobus Marais	15188397
Rikard Schouwstra	15012299
Victor Twigge	10376802

Stakeholders

Computer Science Department of University of Pretoria:	Vreda Pieterse
eCivix	Daniël Eloff Chairperson

Contents

1 Introduction

We Cerebero are working together with eCivix to create a new web based game. The idea of the game is to create an election simulator to teach High school students how elections work and what their vote essentially means in the greater scheme. The user will create party that they will control. The game will revolve around the party gaining funds and man power to do campaigns and gain more funds and man power to run bigger and more effective campaigns. The user with the score at the end of the game wins. The user will be playing against an AI (Artificial Intelligence) player that we will program. The AI will try be more effective/ more successful than the user. There will be a leader-board with all the users' scores and at the end of the client's event a winner will be chosen for a prize.

2 Functional Requirements

2.1 Game Play

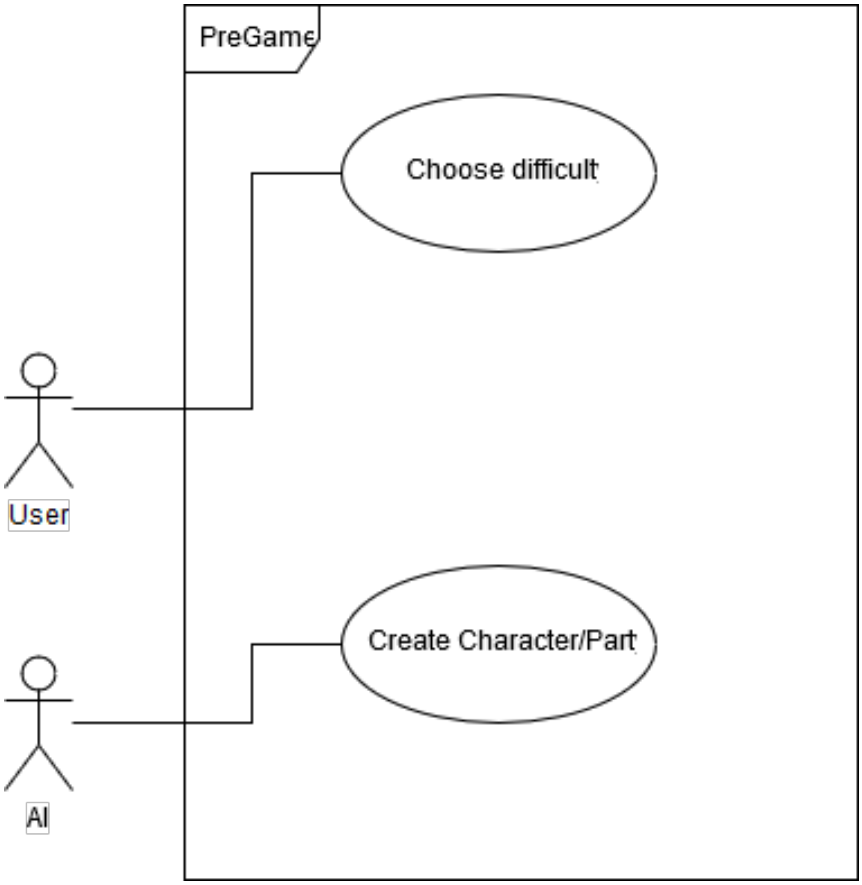


Figure 1: Pre game Use Case diagram

- Create User/Party
  - Select and Customise the person/party that the player will be in the game
  - Preconditions
    - Registered and logged into their account
  - Post-conditions
    - Started the game

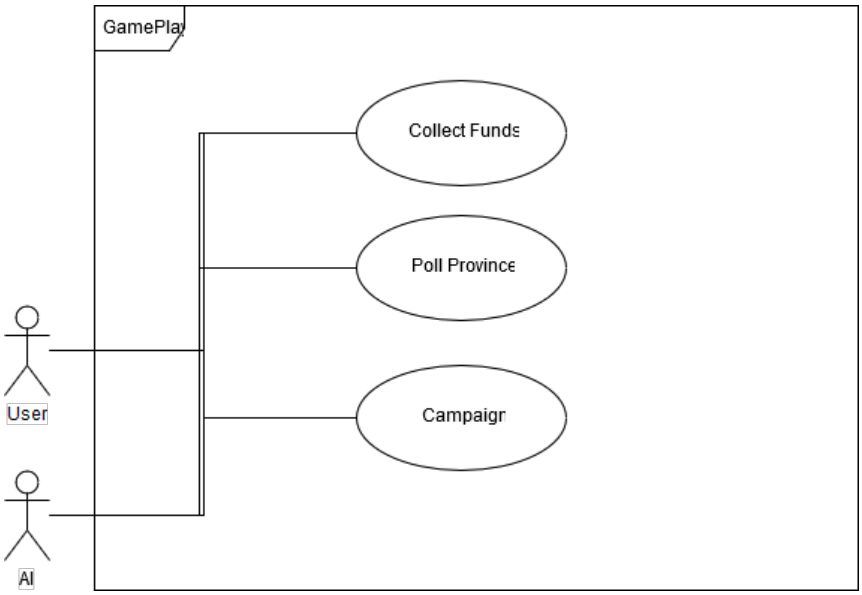


Figure 2: Game play Use Case diagram

- Start Fund raising
  - Start collecting funds nationally to campaign provincially
  - Preconditions
    - User turn started.
  - Post-conditions
    - Got funds to campaign
- Polling
  - Poll to check how good support is in that province
  - Preconditions
    - Have enough national funds to poll in the province.
  - Post-conditions
    - See more or less how many supporters you have in that province.
- Campaigning
  - Campaign in a province to gather support
  - Preconditions
    - Have enough national funds to campaign in the province.

        * The province should have been polled.
    - Post-conditions
        * Gather or lose supporters based on how your campaign went.

- Ending turn
    - Ends the turn and allows AI to play their round
    - Preconditions
        * User is satisfied with what they have done in their turn
    - Post-conditions
        * AI gets their turn.

- Ending game
    - The game has ended.
    - Preconditions
        * The date of the election has arrived.
    - Post-conditions
        * The winner of the election is announced.

- Awarding man power for an action
    - Users are awarded man power for doing campaigns
    - Preconditions
        * A campaign was completed before the end of the game
    - Post-conditions
        * Based on user's party preferences the effectiveness of the campaign is calculated.
        * The calculated value is then added to the user's total man power.

- Artificial Intelligence
    - An intelligent agent that can play against the user to try and win the election
    - Preconditions
        * Game started
    - Post-conditions
        * Game ended

- Leader-board
    - Display the user's position on the leader-board.
    - Preconditions
        * The game has ended.
    - Post-conditions
        * User was able to view his position on the leader-board.

2.2 User management

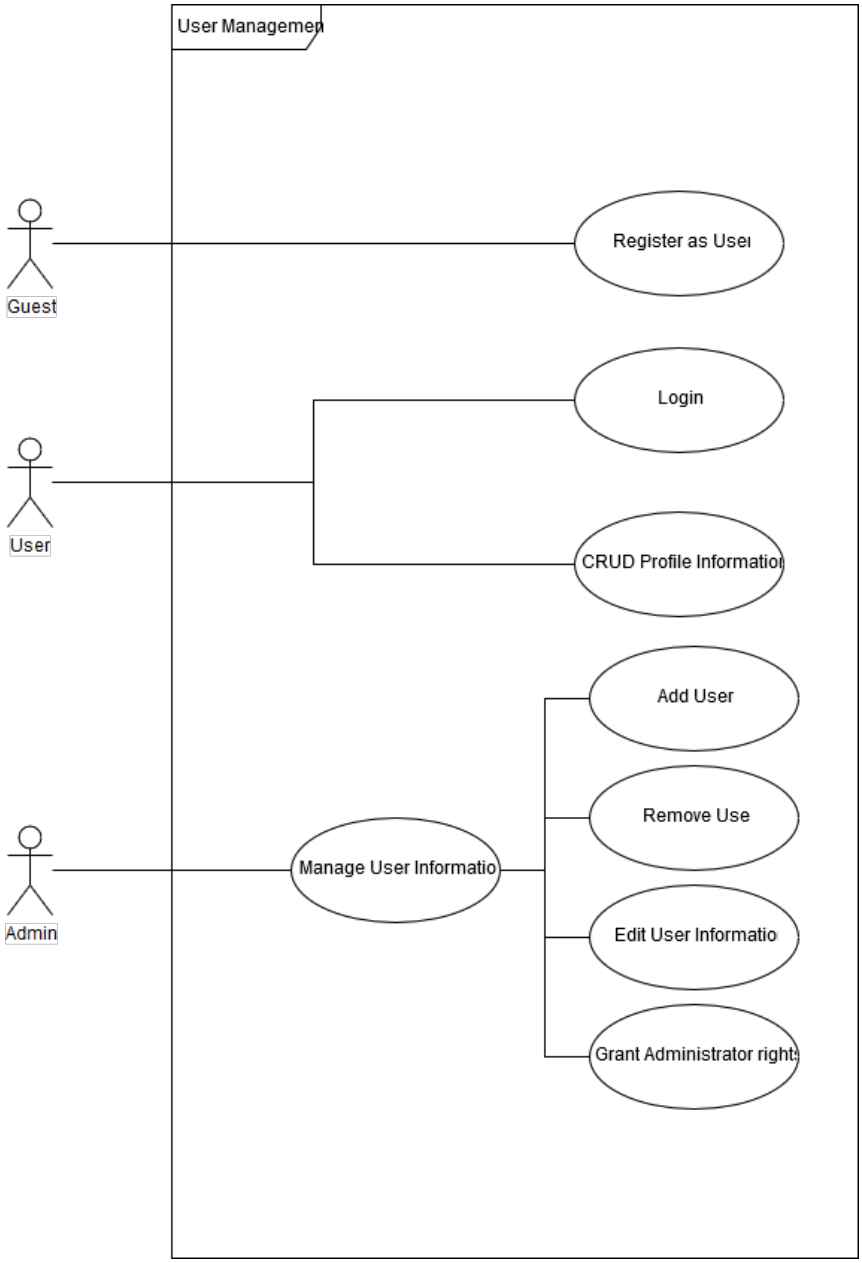


Figure 3: User management Use Case diagram

- Register as user or admin
  - Use your personal details to register on system in order to gain access to more of the applications features.
  - Preconditions
    - Submit details on registration form
  - Post-conditions
    - A personal account and profile is created for the user on the system
- Login

  - Query user's valid details on the system's database through the login page to gain access to the account.
  - Preconditions
    * Have to be registered on eCivix Election Simulator
    * Have to submit his/her correct details on the login page
  - Post-conditions
    * The user is redirected to their profile
    * Have access to user features

- Admin manage user accounts
  - User management is necessary if a user experiences problems with their password or any account related issues.
  - Preconditions
    * Have admin account and therefore rights
    * Have internet access
    * A management system in place
  - Post-conditions
    * Users' account related problems can be solved

- Edit profile information
  - Allow users to add a summary and personal information. Also create provision for the use of a username for anonymity.
  - Preconditions
    * Submit information to the server
  - Post-conditions
    * Profile details will be changed to his/her new details

3 Architectural Specifications

3.1 Architectural Requirements

- (Given to us by the client: eCivix)
- Web-based
  - The web-based game should be interacted with by users through the web.
- Performance
  - The game must be optimised to function and calculate each decision made by the user in the fastest possible way.
- Privacy
  - Legally speaking all information captured by the web-based games registration and login process has to be POPI compliant. The team will be assisted by the legal team of eCivix to ensure that all requirements in terms of privacy are met.
- Security
  - This project will need to be able to keep user data secure and prevent illicit access by third parties.

  - Scalability
    * The project is intended for use as an online service, with potentially many different types of game-play and play-throughs. The project should scale as the size of the game itself grows.

- Game Theory
  - The end game should include mathematical models that simulate the conflict and cooperation between intelligent and rational decision-makers to showcase and represent behavioural patterns of South African voters. In addition, AI players should employ game theory to determine their optimal course of action in any given circumstance.

3.2 Quality Requirements

- Robustness
  - The game should have the ability to cope with errors during execution and cope with erroneous input.
- Maintainability
  - The game should be easily maintainable (in good condition), to make future maintenance easier.

4 Integration Requirement

4.1 Hardware

Minimal hardware requirements

- Client/ user side
  - A computer with a browser that supports WebGL
  - User must have access to the Internet
- Developer side
  - A server that can host a database
  - A server that can handle high volumes of requests

4.2 Software

- The system will be divided into the following subsystems and components which will communicate with each other via their respective interfaces:
- Authentication
  - An interface that handles login and register requests from users.
- Artificial Intelligence
  - Users will compete against the computer which is driven by AI. The AI subsystem needs to monitor the user's decisions and also be able to analyze data from the pre-populated database consisting of political data. The AI will use this information to make moves and decisions dynamically.
- Users

  - The management of users from the admin side via CRUD methods.
- Access Module
  - The web interface that the users will interact with.
- Data
  - This subsystem will retrieve information from the database, process it and send it to the requester.
- Gameplay
  - A subsystem that provides dynamic logic for the game. It consists of the following: user and AI decisions, game rules, political data etc.

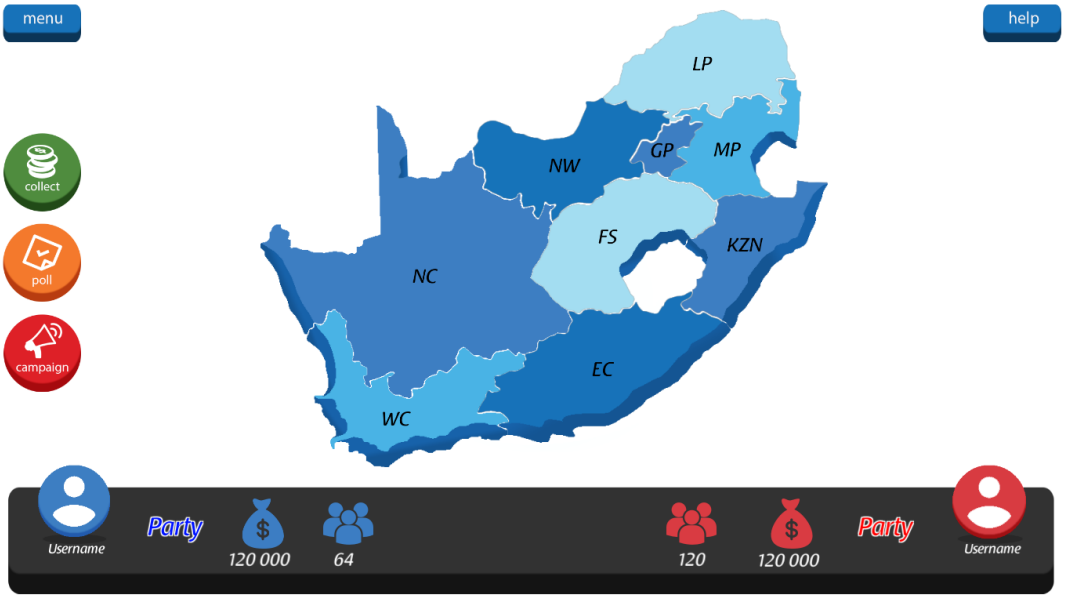
5 Constraints

5.1 Legal constraints

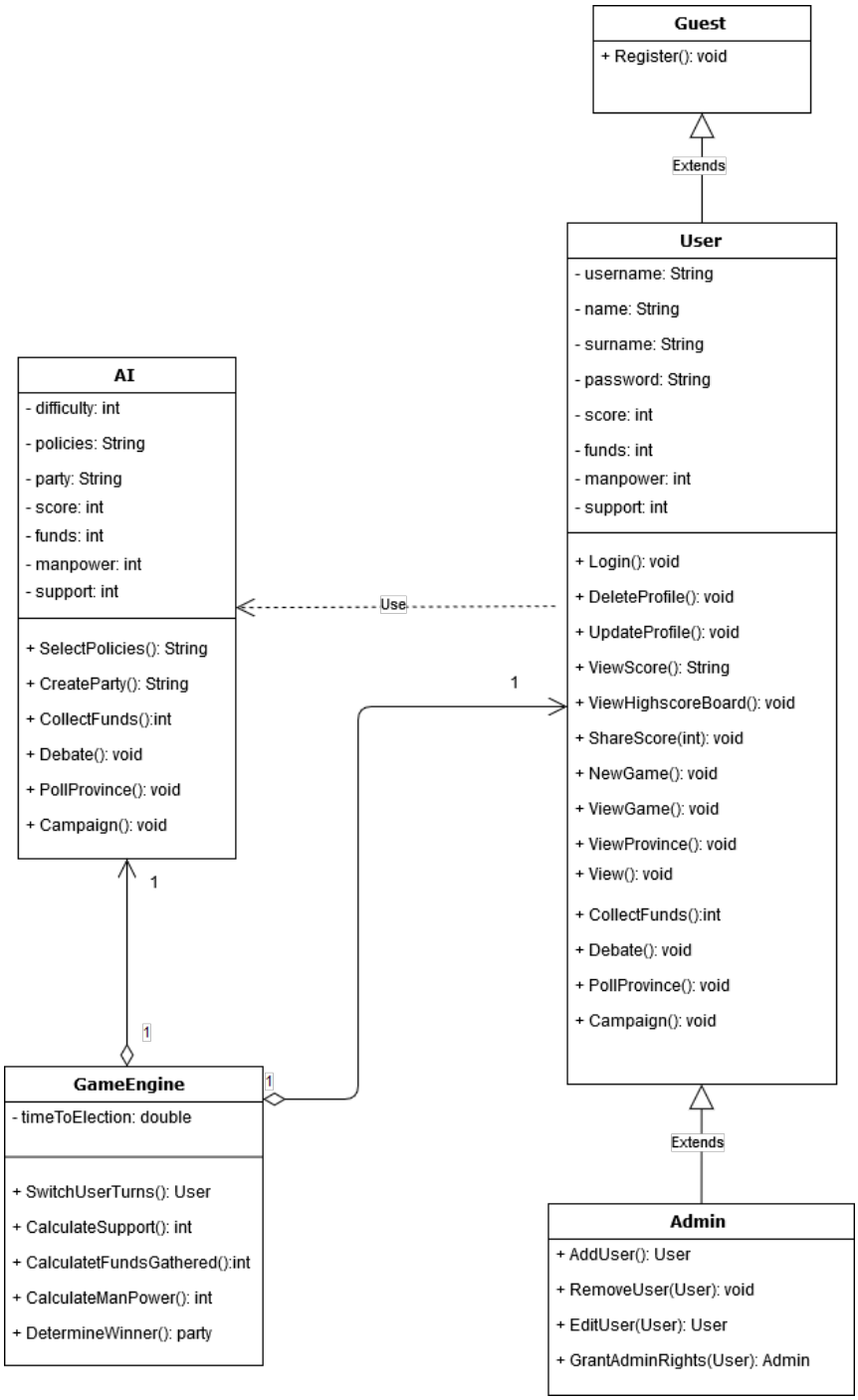
- User information such as emails is sensitive and must be protected.
- A private repository would need to be used if user emails are used to register the users.

6 Appendix

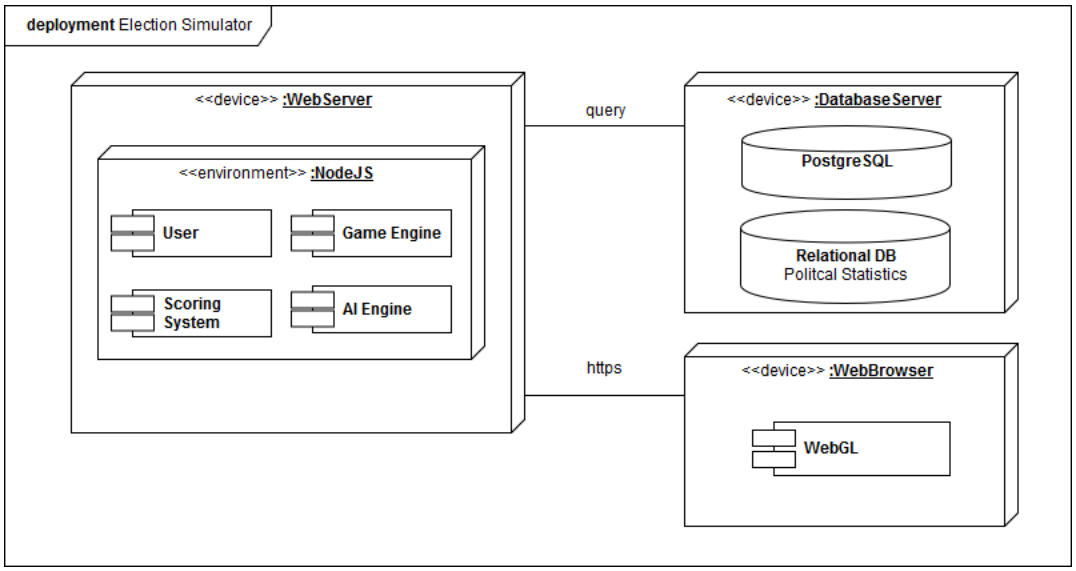
6.1 GUI illustration



6.2 Domain Model



6.3 Deployment Diagram



6.4 Github

Github

6.5 Trello

Trello

6.6 Swagger

Swagger

6.7 Heroku

Heroku

6.8 Travis CI

Travis CI