DESIGN DOCUMENT
D-RISK

TEAM INFERNO

Table of Contents

1. INTRODUCTION

I. Project Overview

Risk is a turn-based board game, in which 2 to 6 players fight over the occupation of 42 territories in a political map, trying to achieve a secret mission that generally requires the control of the territories in the map.

The goal of this project is to develop a turn based client server Risk game, in which users play the game through a web site, and that offers high configurability, in particular: it allows users to define and upload map configurations, including not only the aesthetic aspects such as images, etc., but also specific numbers of territories and neighboring relations (and optionally stronger restrictions on the number of players); it allows for maps to automatically adjust granularity, by collapsing neighboring territories according to number of players and a user selected complexity for the game (e.g., easy, medium, hard).

II. Scope

This document describes the implementation details of D-Risk game described in the requirement specifications. It gives an overview of all the development phases, the methodology used, details about various diagrams, and also screen shots of the various HTML pages.

III. Sprints

The project development phase was divided in total of four main sprints and one mini sprint.

Sprint I (Sep 21 – Oct 5):
This was the start of the development phase. In this phase each team member was given the task to make them familiar with the different aspects of the game i.e. the learning the game, build a model for development of each of the functionality, gather data about the requirements and the approaches to start the development of the game. After getting familiar and gathering required data, the team members drafted a structure as to how the development of the project will proceed. This phase was the most important as any crucial requirement missed at the beginning could have affected the development at later stages. We prepared the basic HTML pages and the gathered all the required tools that would be used in the later development stages. We were ready with tools like Adobe illustrator for generating the SVG data for creation of maps, an amazon web server to launch our website. We held short sprint meetings so that everyone was in sync with the ongoing development phase.

Sprint II (Oct 6 – Oct 19):
In this sprint, we subdivided the task into groups. Two of us worked on generating the default map (large size) SVG data using the adobe illustrator tool, other two worked on making the HTML files such as login/signup, and one guy was given the task to learn ‘raphael js’ which was used to draw the map and add animations to it. By the end of this sprint we had our map ready with animations added to it. Simultaneously we also kept track of how the functionalities will be added to it such as the different phases of the game (deploy, attack, execute). We also worked on maintaining the google sites as well by adding documents and other minor details which was used seen during each development phase. This was used as a tracker of our development activities.

Sprint III (Oct 20 – Nov 2):
This sprint was mainly focused upon making the functionalities for two players so that they can perform all the phases of the game without any errors and also launching the game on the amazon web server. In this sprint we also learned about node JS which was our server and this was the backbone for data sharing between different clients. Focusing on data sharing was also one of the most important part of this sprint as all the client has to be in sync with each other client. Once data sharing was proper between two clients we discussed to extend the functionality for more players.

Sprint IV (Nov 3 – Nov 16):
This was the last sprint of the project development phase. In this phase we extended the functionalities for 6 players and also added 2 more maps i.e. medium and small. The idea behind adding different complexity level map was to give the users a choice to choose different size map in case there are less number of player to the start the game. In this phase we also added animations and worked on creation of more html pages to make the site more detailed giving details about the game.

Stabilization Sprint (Nov 17 - Nov 23):
In this sprint the developers of the game themselves played the game from user's perspective and kept track of the bugs which were found. The main focus of this sprint was to fix all the minor bugs that were found and fix them. Each team member worked towards stabilizing the whole product. Simultaneously we worked on finalizing the documents and preparation of the presentation.

2. ARCHITECTURE DESIGN

I. FRONTEND

Since this is a web based project, the web pages are designed using HTML, CSS and JS. The major part of the client programming is done in JS. All the logic for the functionalities are written in java script. The game has one starting page where the users are given the option to login, signup and other html files to check out the details of the game.
Once the user logins he moves to the next html page where he/she has the option to create a game or join an existing game is the game Id is known. The functionalities given at this page include creation of a new game by entering the e-mail id of players (up to 5), selecting the complexity of the game (easy, medium, and large), and if a game ID is known then joining the existing game by entering that game ID.
Once the above details are entered and all the players have joined, the host can start the game by clicking the start game button. After this all players are moved to the next page where they are assigned different territories as per the selected map.

II. BACKEND

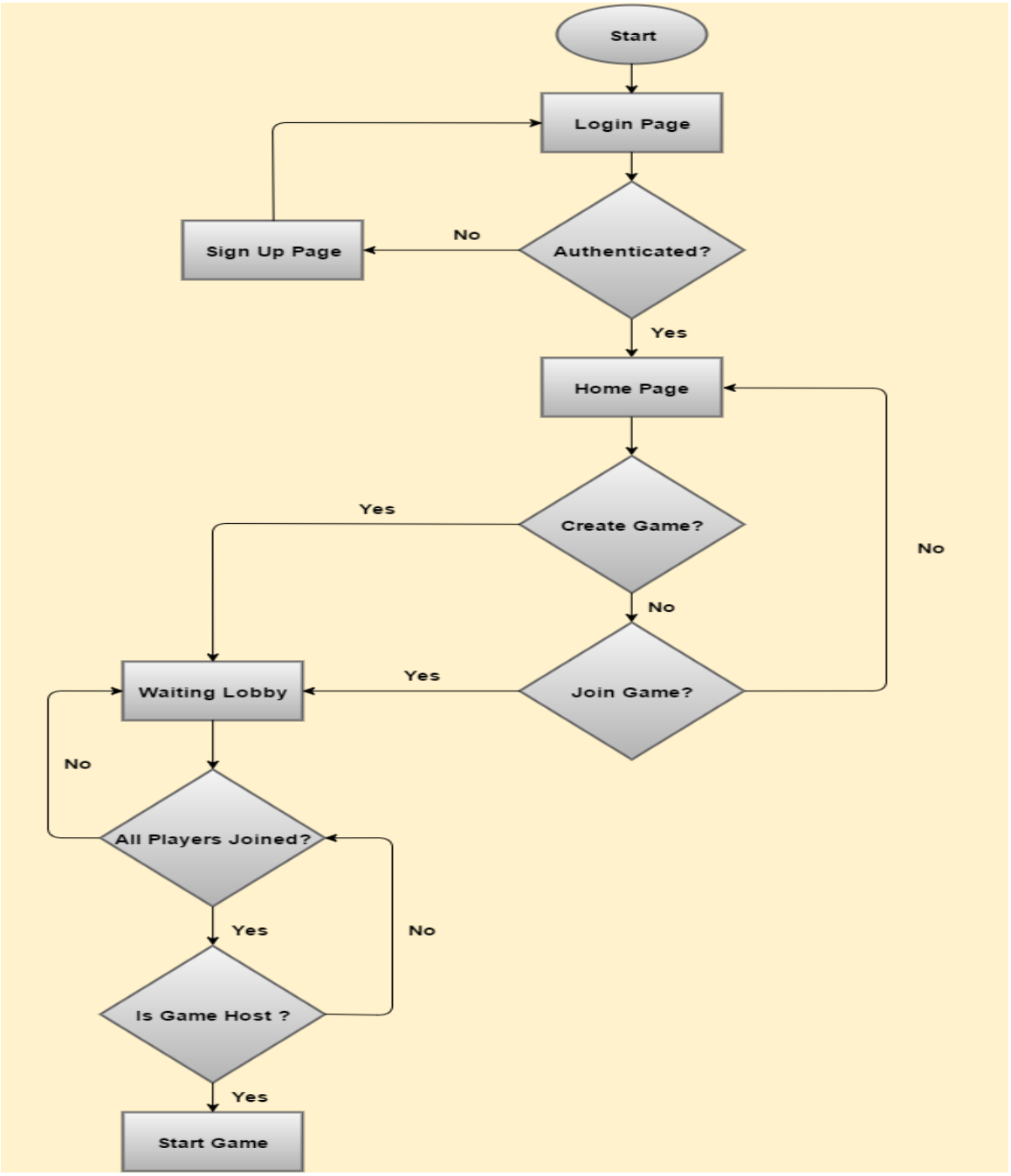
The server used for hosting the website is a Linux based server provided by Amazon web services. We have one instance of the server running which can handle multiple request for multiple games. We have used Node JS to handle the communication between different clients. The actions performed by one of the client are emitted to all other client who is playing.

III. DATABASE

We have used mongo DB to store the details of all the users and other required details which might be used in between of the game.

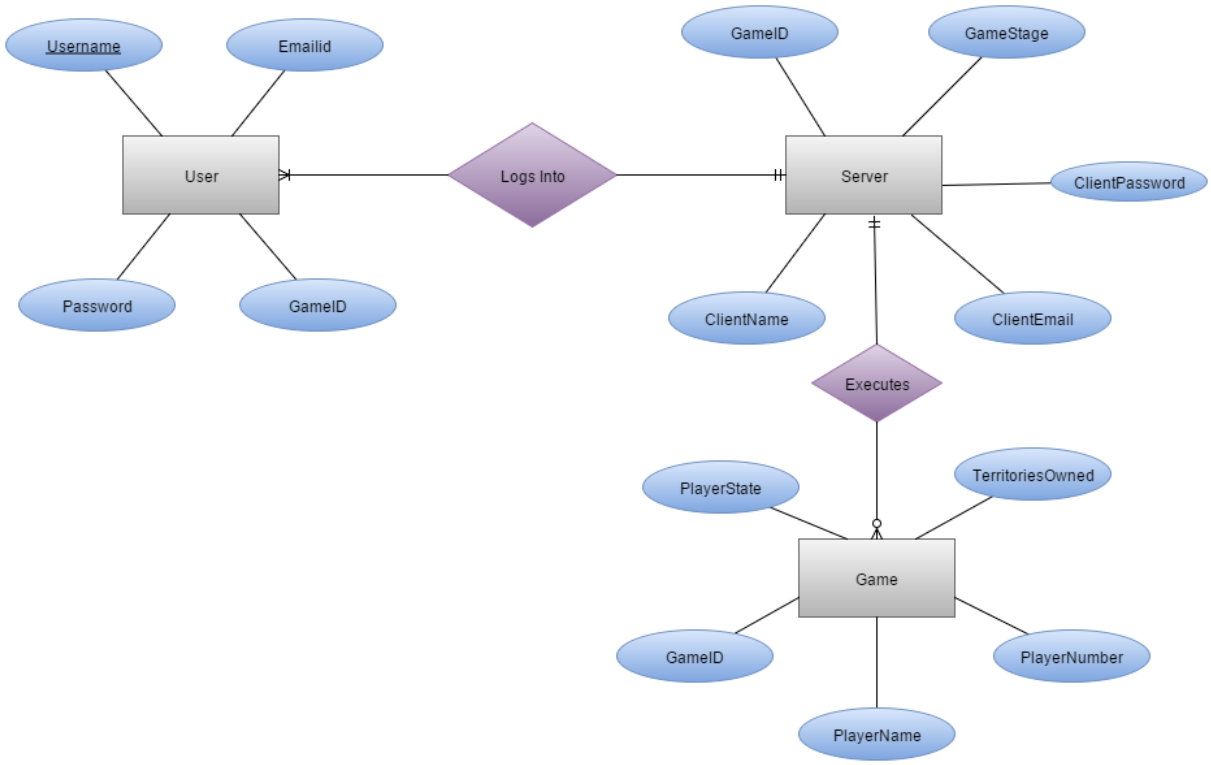
3. SYSTEM OVERVIEW

I. FLOW CHART



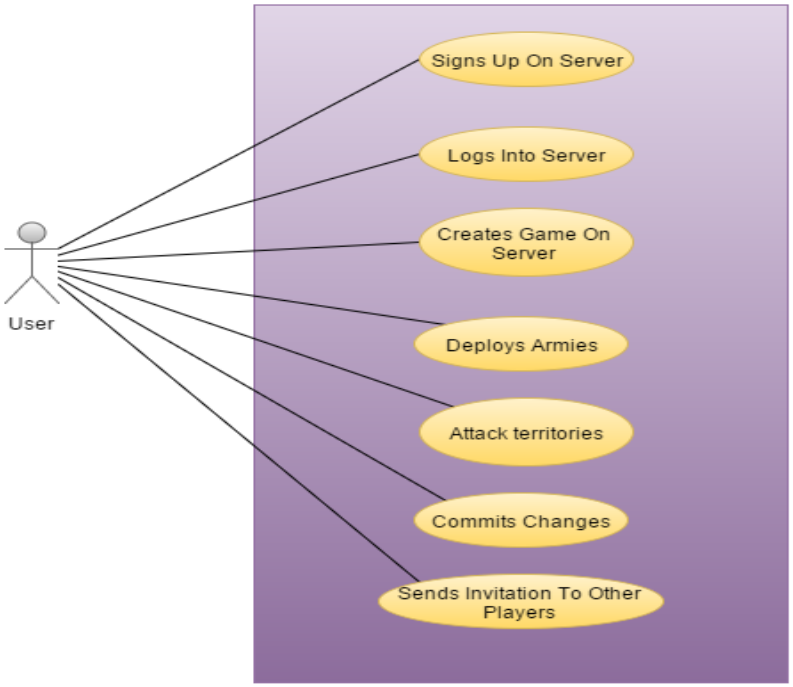
This Flow chart describes the flow from the state where the user opens the DRisk webpage and till the Game is loaded. The user has option to login or signup and then Game options to create a new game or join an already existing game. If the user creates a game then he/she will be the host for the game and have the option to start the game when all the players have joined the game room.

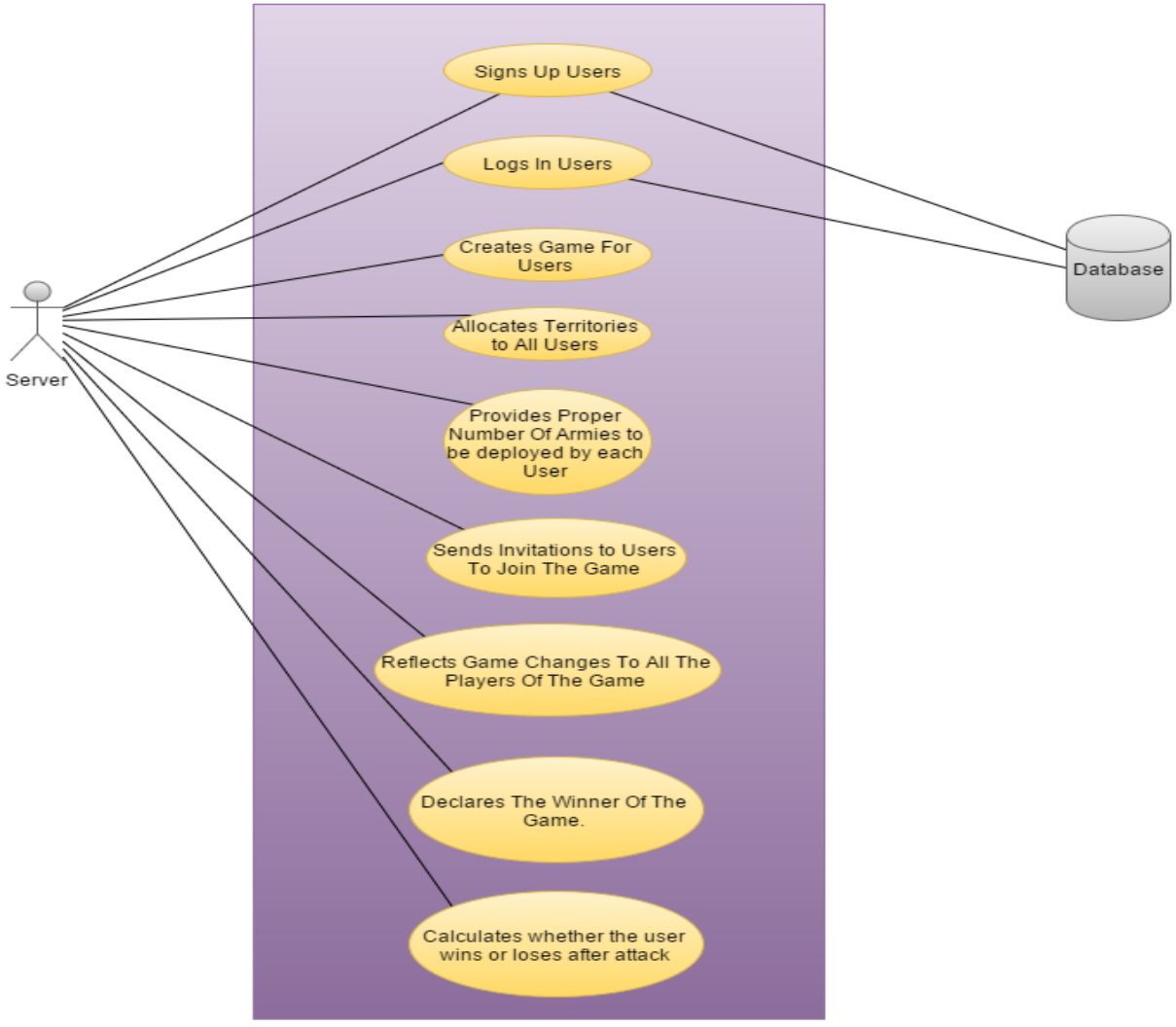
II. ER DIAGRAM



This diagram shows how the client logs in to the server which in turn launches the game. There are three entities viz. User, Server, and Game. The user logs on to the server with Name, Email Id and password. The server then accepts the request for hosting or joining a given game with a given game ID. Once all users join the lobby, the game is then launched.

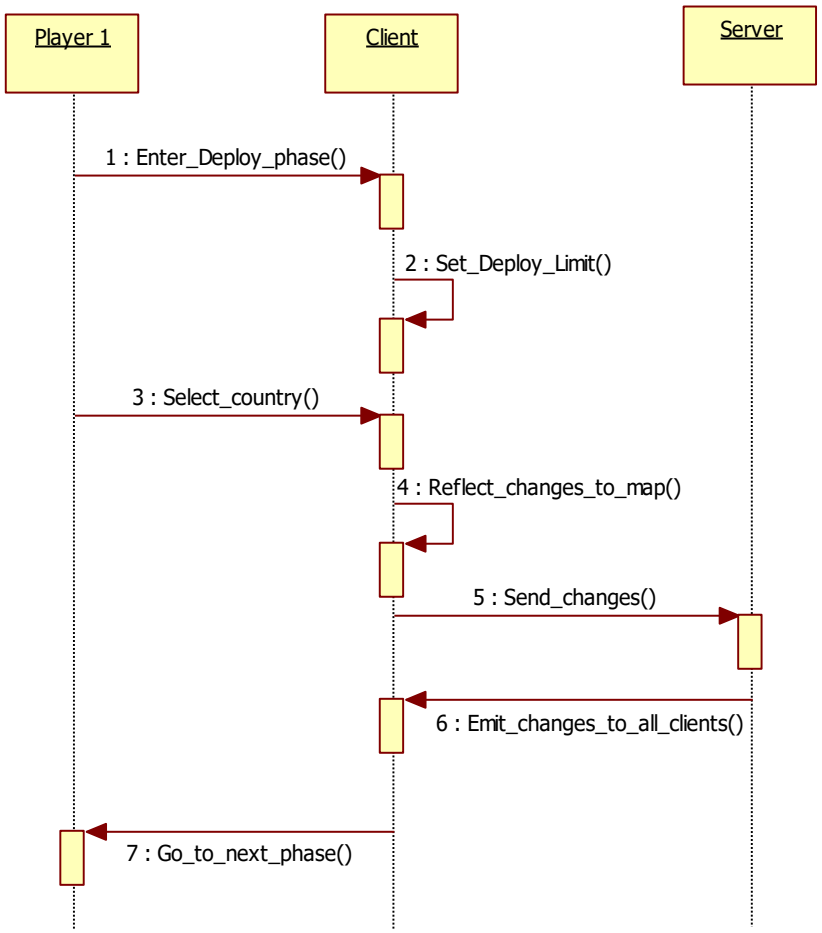
III. USE CASE DIAGRAM

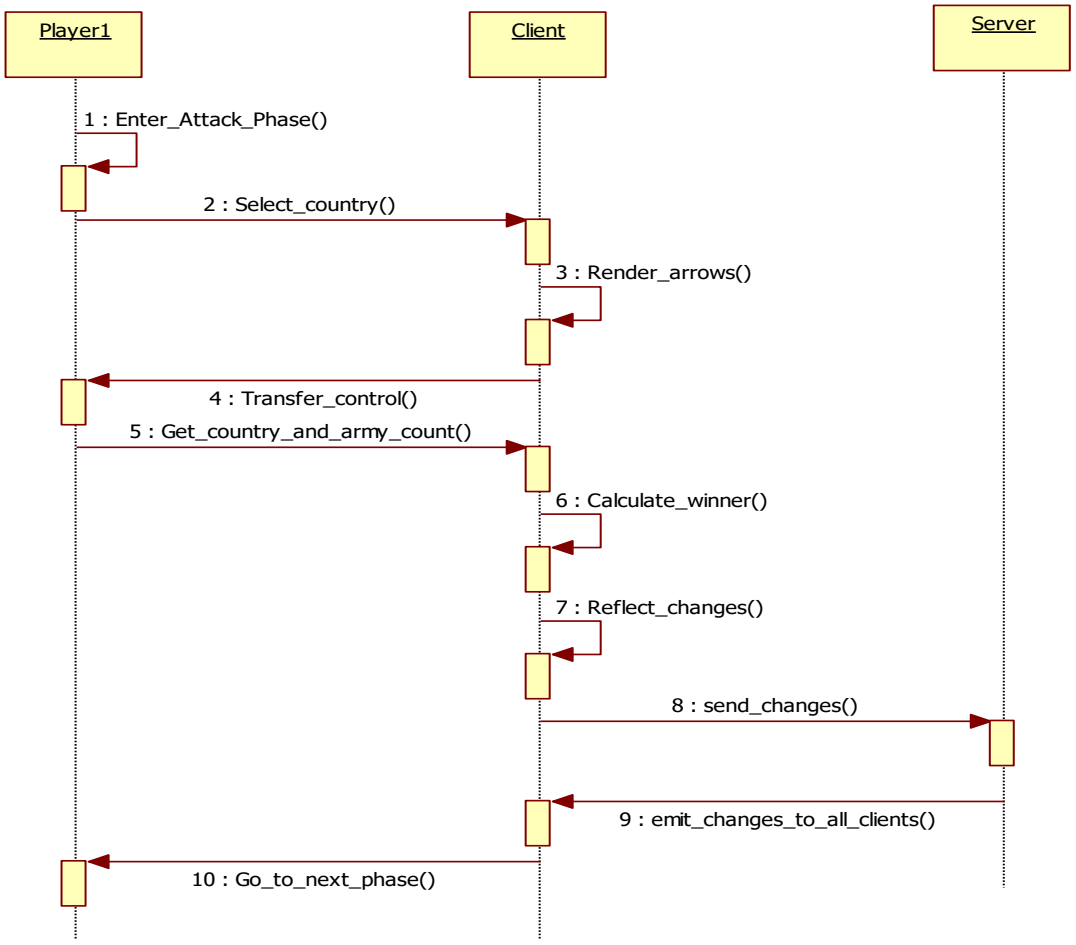


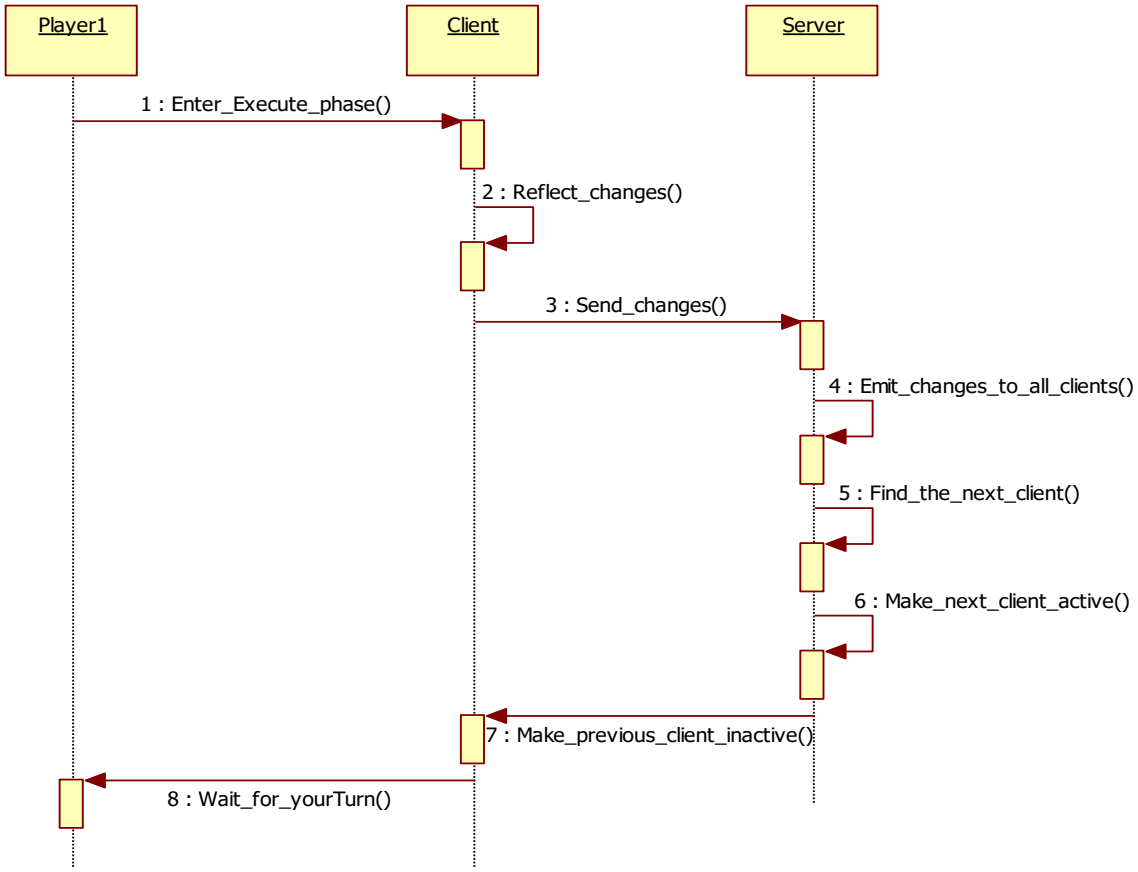


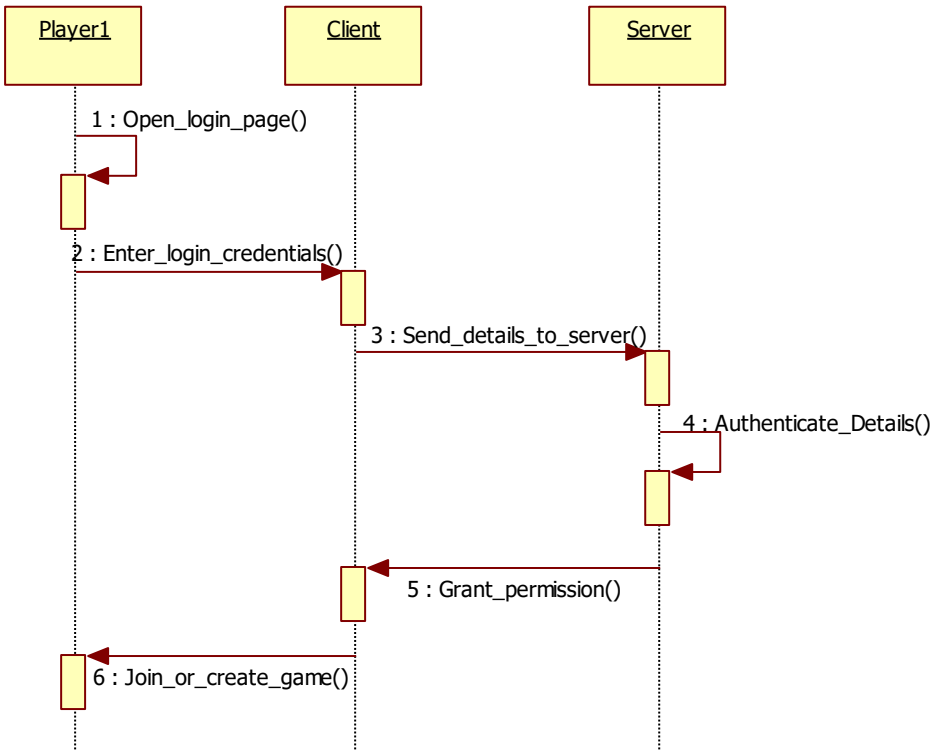
The above diagram shows the Use case Diagram for the user and the server. The user is responsible for signing up, logging in, and creating game on server. He can make his moves of deploy, attack and commit and also can invite other players to join the game via their email ID through the server. On the other hand the server is responsible for creating account of the user on the database and signing him in when required. The server creates a game for the user and allocates a game ID for the game which is sent to other users when the user wishes to invite them to play the game. Also the server is responsible for checking if the attack of the user was a successful attack or not. The server keeps checks of the number of armies to be deployed by each user as well and if any user acquires all the territories, the server declares that user as the winner of the game.

IV. SEQUENCE DIAGRAM






Player1
Client
Server
1 : Enter_Execute_phase()
2 : Reflect_changes()
3 : Send_changes()
4 : Emit_changes_to_all_clients()
5 : Find_the_next_client()
6 : Make_next_client_active()
7 : Make_previous_client_inactive()
8 : Wait_for_yourTurn()


Player1
Client
Server
1 : Open_login_page()
2 : Enter_login_credentials()
3 : Send_details_to_server()
4 : Authenticate_Details()
5 : Grant_permission()
6 : Join_or_create_game()

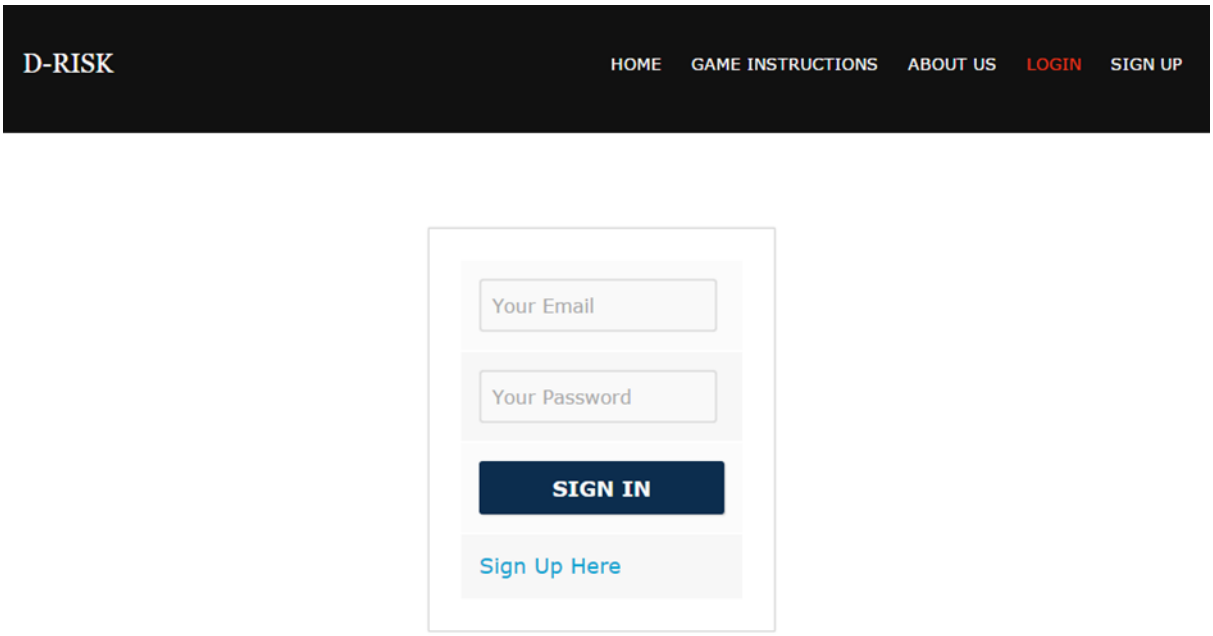
4. HUMAN INTERFACE DESIGN

I. OVERVIEW OF USER INTERFACE

In this section we will present all the web pages that will be displayed on our website. The primary goal is to help user understand all the features and actions that are developed as a part of this game. We will provide a detailed description with images for each of the pages from the start i.e. the home page to the end page in the following sections.

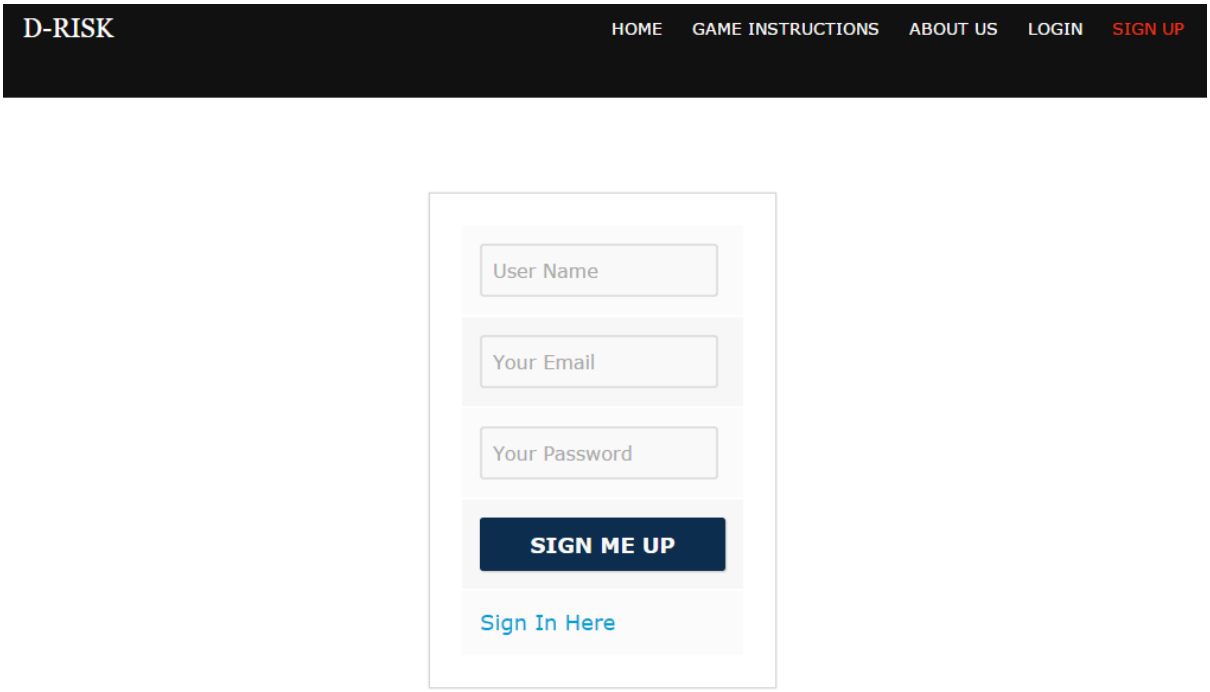
II. SCREEN IMAGES

Login Page



Control Type	Name	Description
Textbox	Email ID	Enter the Email ID
Textbox	Password	Enter user's password
Button	Sign In	Click on Sign in to advance to the next screen and begin using application
Button	Sign Up Here	Users who have not previously registered for the DRisk Web Application must click on this option to access the "**New User Sign Up**" page.

New User Sign Up



Control Type	Name	Description
Textbox	User Name	Enter the Username/Nickname
Textbox	Email	Enter the user's e-mail address. E-mail addresses are not case sensitive.
Textbox	Password	Passwords must be at least eight characters long and contain at least one letter character and one numeric character. Passwords are case sensitive.
Button	Sign Me Up	Select Sign Me up to submit the registration information and continue to next screen.
Button	Sign In Here	Click on Sign in Here to move to the login page screen and begin using application.

Home.html


Nickname:
Create Game
Already have game id, Enter Game id !
Nickname:
Game Id:
Join Game

Control Type	Name	Description
Textbox	Nickname	Enter the Username/Nickname
Button	Create Game	Click on Create Game to start a new Game as the user being the host of the Game.
Textbox	Game ID	Enter the Game ID, if you wish to join an existing game.
Button	Join Game	Click on join game to advance to the next screen.

Login as Game Host

Control Type	**Name**	**Description**
Textbox	Email Invites	Enter the Email ID's of the players to send invite.(Maximum 5)
Drop Down list	Complexity Level	Select the Complexity level of the Game you want to play. The options are according to the Map size as "Easy", "Medium" and "Hard".
Button	Join Game	Click on join game to advance to the next screen.

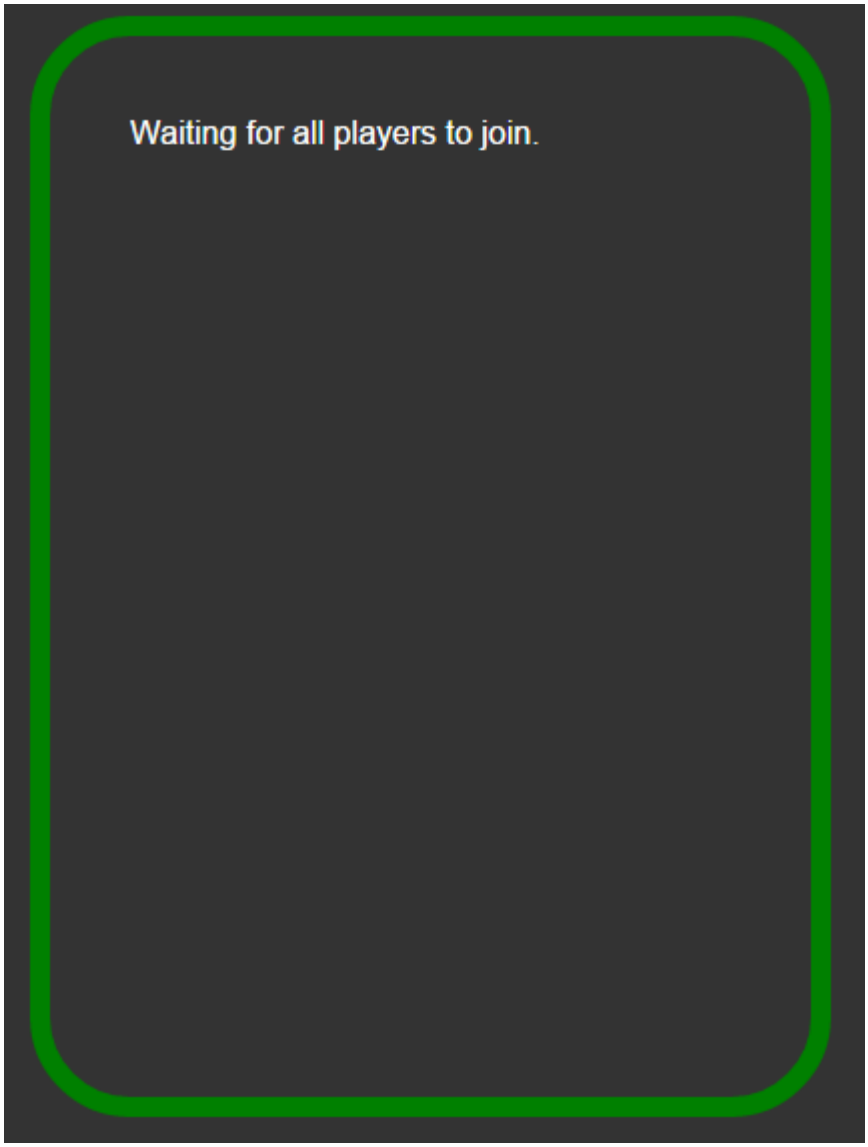
Start Game Option for Host

Control Type	Name	Description
Button	Start Game	Click on Start game to begin playing DRisk when all the players have joined.

Join a Multiplayer Game

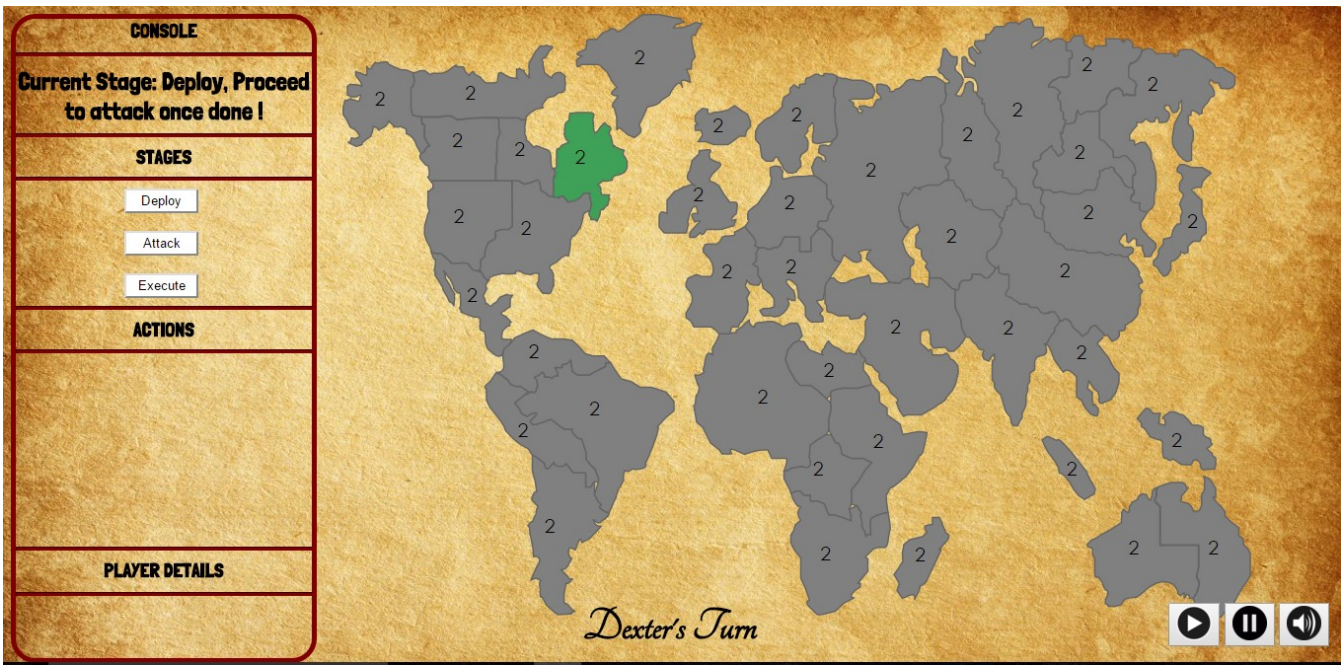
Control Type	Name	Description
Textbox	Nickname	Enter the Username/Nickname
Textbox	Game ID	Enter the Game ID, if you wish to join an existing game.
Button	Join ID	Click on join game to advance to the next screen.

Waiting Lobby



Control Type	Name	Description
No Control	Waiting Lobby	All the players will be displayed this screen until the host chooses to start the Game.

Game Page



Control Type	Name	Description
Button	Deploy	Click on Deploy to start deploying your armies when it's your turn.
Button	Attack	After Deploying the armies click on attack button to conquer the neighboring territories.
Button	Execute	Click on Execute button when you are finished attacking other territories.

5. TOOLS AND TECHNIQUE

I. Software Development Process

We have adopted SCRUM methodology for our software development process.

II. Google Site

We have maintained our google site to keep a track for the development phase. We have attached all the necessary documents viz. Requirement Document, Design Document, and User Manual. We also have a section which gives detail about the each sprint coverage. By the end of main sprints we have a tab which gives detail about the bugs found during the final testing phase.

Link: https://sites.google.com/a/asu.edu/inferno/

III. Project Management Tool

We have used "taiga" as our project Management tool. Taiga is a project management platform for agile developers. It served our purpose for scrum board. All the sprint details, the user stories and task for the particular sprints were updated over here and tracked accordingly. We have a total of four sprints. The screenshot for each can be found in the google site as well.

Link: https://tree.taiga.io/project/ser515asu-drisk-inferno/

IV. Code Repository

We have used GitHub for our software code management. All the team members contributed to the repository with the code of the functionalities they developed.

Link: https://github.com/ser515asu/DRisk-Inferno