

## Software requirements specification

# WormGame

Version 2.0

Karelia	TIKO	LTP7024 Testausmenetelmät
Author: Juhani Pirinen		Printed: 2nd oct 2015
Distribution: Teacher of the course, Public in GitHub		
Status of the document: draft		Edited: 2nd oct 2015

## VERSION HISTORY

Version	Date	Authors	Description
1.0	13th sep 2015	The group 6	<a href="http://qcheni.github.io/ola/reve/pirkk&lt;br/&gt;aplus/#/">http://qcheni.github.io/ola/reve/pirkk aplus/#/</a> (in finnish)
2.0	2nd oct 2015	Juhani Pirinen	Draft (in english)

## 1. PREFACE

Purpose of this document is to be a very light weight systems requirements specification for a multi-player worm game made in network application development and software testing courses in university of applied sciences. Only those parts of software, that will be tested on testing course, are documented a little more in detail.

## 2. SPECIFICATION

### 1.1 Description

This is a multi-player online version of the classic worm game, that will run in web browser. There can be maximum 4 worms at the same time on game board. Each player controls his worm. Food for worms is randomly spawned to the game board. When worm eats food, its length grows, and thereby worm gets harder to control. Worm dies, if it collides game board walls or other worms. Each eaten food will increase player's score. Winner of the game is player, who has highest score.

## 1.2 Architecture

This will be a server-client application. Game server runs the game, when clients' role is to be only a controlling dashboard interface to it through web browser.

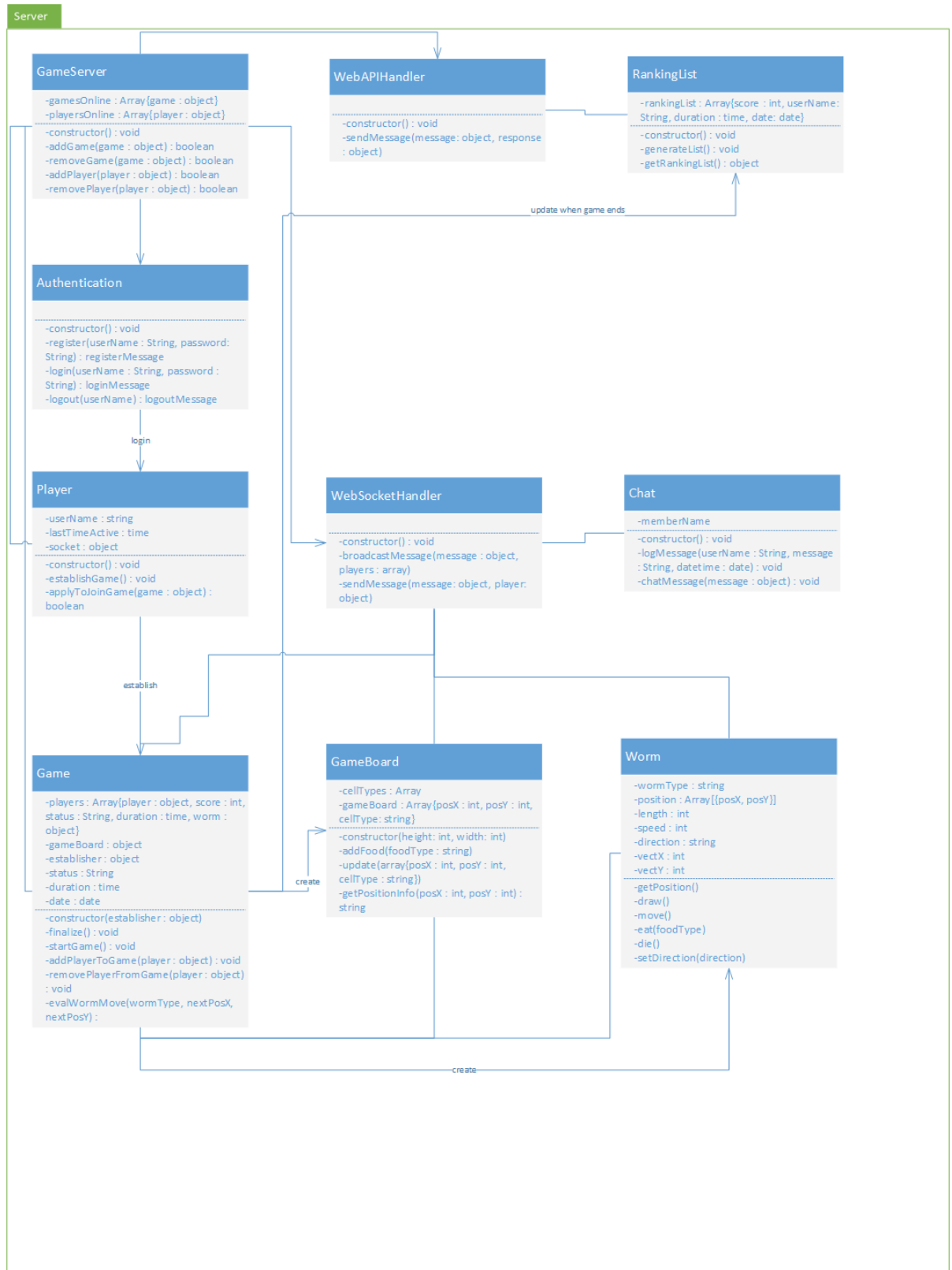
Server:

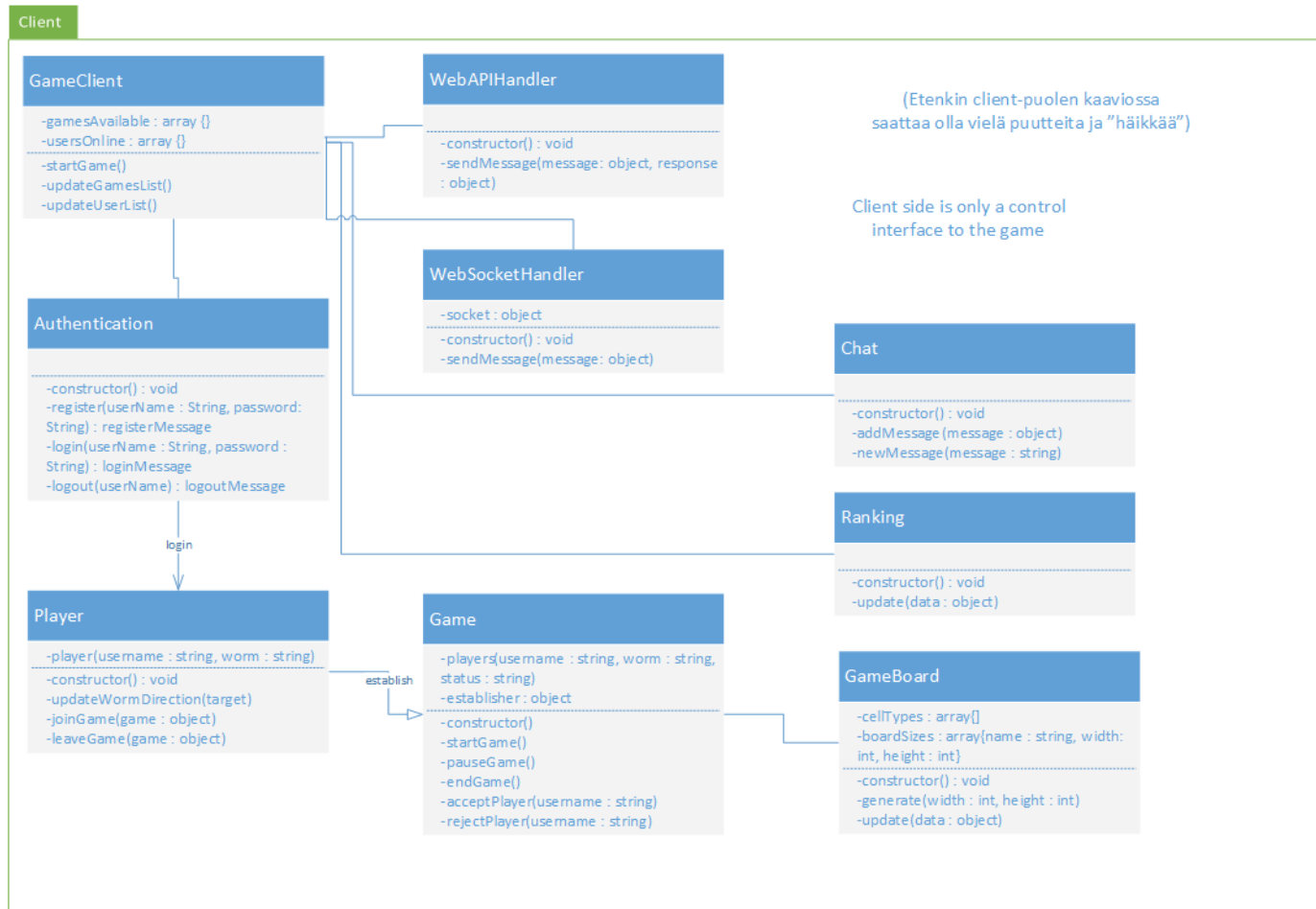
- Node.js framework
- Express.js framework
- MySQL database
- Socket.IO for utilizing web sockets

Client:

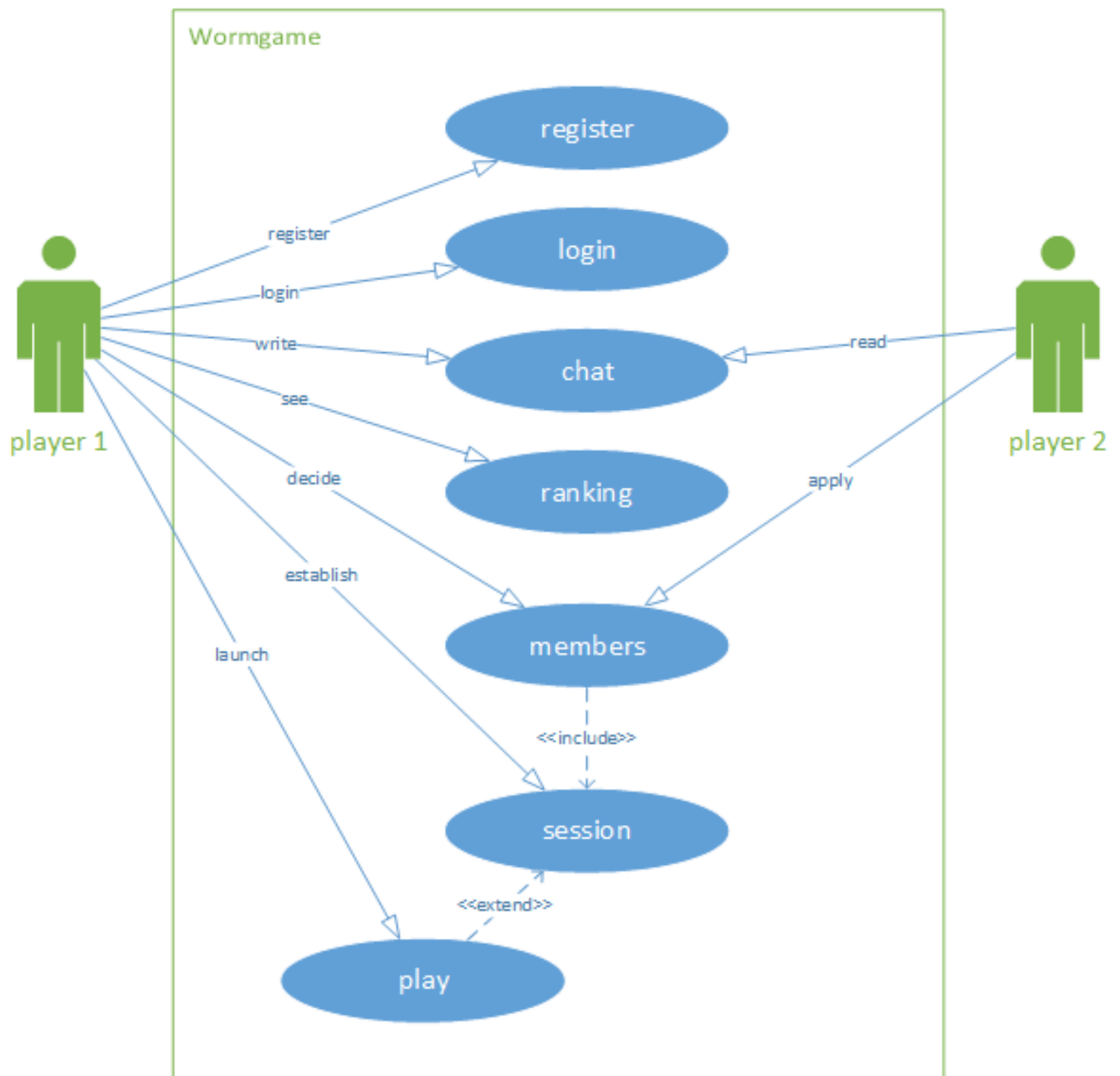
- HTML5 + CSS3
- Javascript (ecmascript 5)
- Socket.IO for utilizing web sockets

### 1.3 Concept models





## 1.4 Use cases



## 1.5 Functional requirements

- Single-player online-game for registered users
- Multi-player online-game for registered users; max 4 online players; each has unique worm colour; each player uses own browser windows for the game
- User registration (username and password) open to anyone; returns information if fails
- User authentication (username and password); returns information if fails
- Ranking list of highest scores in the game
- Text chat between registered and unregistered users
- Chat messages archived in the database (user, ip, datetime, message)

## 1.6 Non-functional requirements

- Playable; controlled by mouse or keyboard
- Modular and following OOP paradigm
- Installable to Red Hat OpenShift
- Passwords encryption in database
- Testable



## 1.7 Project plan

### Week 4

- Specification
- GitHub repository
- Single player game (UI/client)
- User registration (UI/client, server, database)

### Week 5

- Authentication (UI/client, server, database)
- Ranking list top 30 (UI/client, server, database)
- Chat (UI/client, server, database)
- Online players (UI/client, server)

### Week 6

- Multi-player game (UI/client, Server, Database)

### Week 7

- Finalizing the project
- Installable to OpenShift
- Software testing (partially)
- Debugging