

Spaghetti-Coders Jungle Sprint 2

EJ Lee
Sam Stobbelaar
Jeff Bradley
Vlad Stepanuga
John Miller



User Stories (overview)

• Register a new account

Acceptance Criteria

- Test with all valid information
- Test with existing username
- Test with existing email
- Test with incorrectly formatted email
- Test with missing fields
- Test with username that includes special characters
- Test with a username that is too long
- Test with a password that is too short
- Test with a password that is too long

• Login to application

Acceptance Criteria

- Test with valid registered user information
- Test with an invalid username
- Test with an invalid password
- Test with unregistered username
- Test with incorrect password
- Test with blank fields
- Test with all invalid fields
- Test with special characters

• Play Jungle

Acceptance Criteria

- Test to see if the game board renders on the site.
- All pieces render on the board before the game starts.
- Test to see if game board is updated once a move is made.
- Test move prevention until opponent makes a move.
- Test rendered elements once game end has been reached.
- Test for JSON object validity once a move has been made.
- JSON object validity on move "locking" status by player.
- Test validity of the logic of each piece's movements.
- Test that end state of match can be reached validly.

From the Spaghetti Coders:
Welcome to the
JUNGLE

Register with a new username, email, and password!

nickname

password

confirm password

email address

Submit

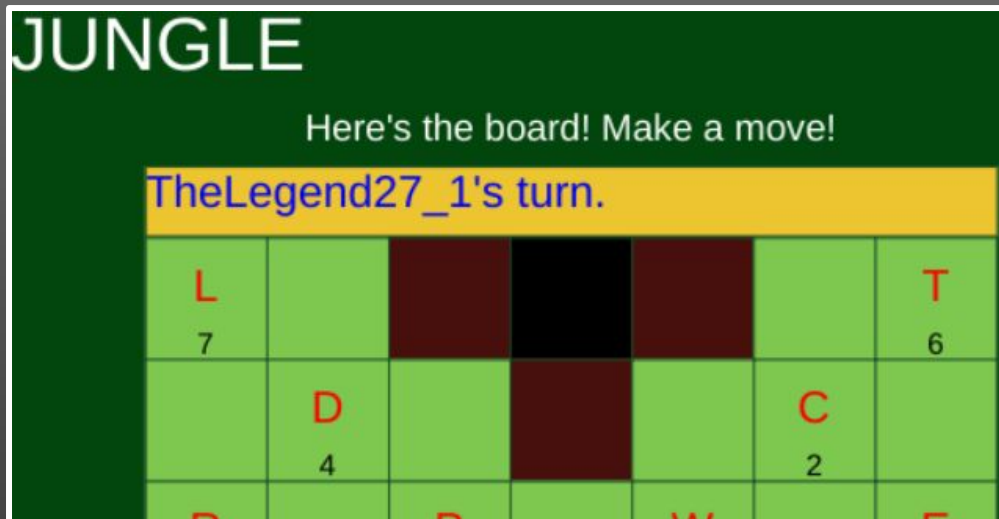
From the Spaghetti Coders:
Welcome to the
JUNGLE

Login and continue playing!

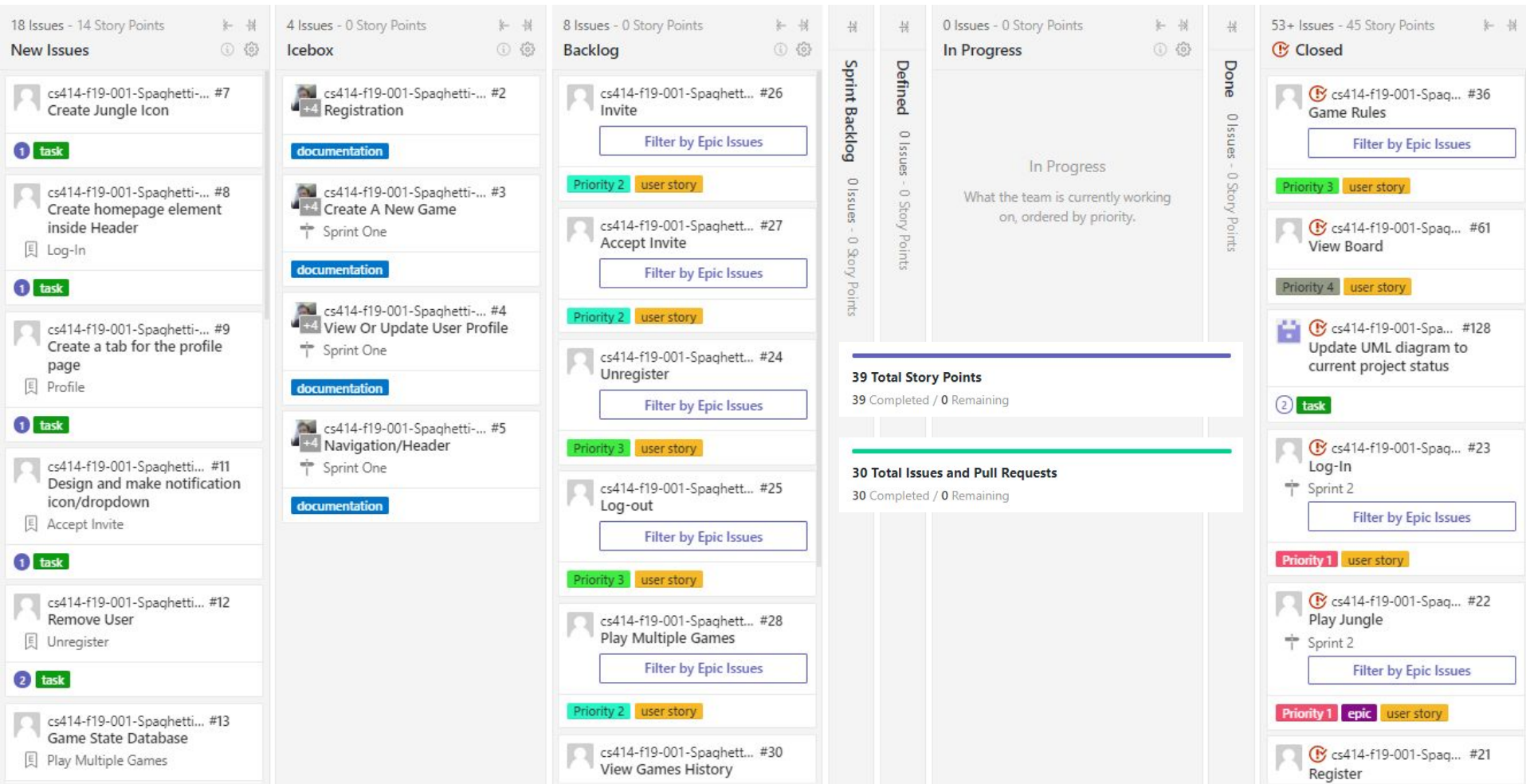
username

password

Login



Kanban Board



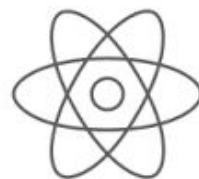
CRC Cards (client)

Application	
<ul style="list-style-type: none">• Handle login• Render Welcome and Main Page	<ul style="list-style-type: none">• Welcome• Main

Login	
<ul style="list-style-type: none">• Validates user credentials• Sends user to main page upon successful login	<ul style="list-style-type: none">• Application• Welcome

Register	
<ul style="list-style-type: none">• Inserts new users into database and creates profile• Validates new credentials are correct format and unused• Sends user to login page after successful registration	<ul style="list-style-type: none">• Application• Welcome

GamePage	
<ul style="list-style-type: none">• Display current state of the board• Show selected piece• Send move made to back end	<ul style="list-style-type: none">• Main• Match



reactstrap

CRC Cards (client & restful api)

Main	
<ul style="list-style-type: none">• Display current and past games from back end• Display current user logged in• Allow user to start a new game	<ul style="list-style-type: none">• Match• Login

Welcome	
<ul style="list-style-type: none">• Initial page user sees upon visiting website• Allows user to go to login or register page• Displays info about Jungle game and devs	

HTTPRestful	
<ul style="list-style-type: none">• Handle register and login requests• Start server	<ul style="list-style-type: none">• Login• Register

InitServer	
<ul style="list-style-type: none">• Initialize the server using an HTTPRestful object	<ul style="list-style-type: none">• HTTPRestful

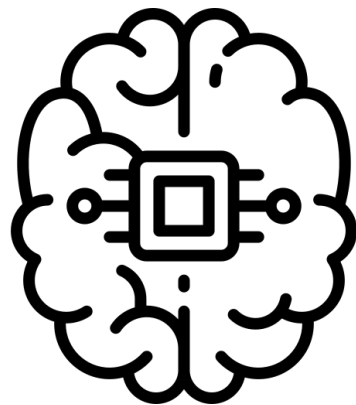


CRC Cards (game logic)

JungleBoard	
<ul style="list-style-type: none">• Create a new board• Initialize the starting pieces• Place pieces• Get piece at certain position• Make moves by updating piece locations• Hold current game state variables for front end communication	<ul style="list-style-type: none">• Piece

Piece	
Rat, Cat, Wolf, Dog, Leopard, BigCat, Lion, Tiger, Elephant	
<ul style="list-style-type: none">• Hold information for each piece like rank, color, row and column• Get and set piece positions• Check and return legal moves for each piece• Check if a move will trap the piece• Check if a move will win the game• Set winner and state of game for front end use	<ul style="list-style-type: none">• jungleBoard• Child classes

Rat	
Piece	
<ul style="list-style-type: none">• Create Rat pieces• Set Rat rank• Check legal positions as Rat has special moves	<ul style="list-style-type: none">• Piece



Game Logic

CRC Cards (pieces)

Collaborators are the same as Elephant

Cat <ul style="list-style-type: none">Create Cat piecesSet Cat rank	BigCat		Piece
	<ul style="list-style-type: none">Check legal moves for Lion and Tiger child classes as they have special moves	<ul style="list-style-type: none">PieceLionTiger	Lion, Tiger
Wolf <ul style="list-style-type: none">Create Wolf piecesSet Wolf rank	Lion		BigCat, Piece
	<ul style="list-style-type: none">Create Lion piecesSet Lion rank	<ul style="list-style-type: none">BigCatPiece	
Dog <ul style="list-style-type: none">Create Dog piecesSet Dog rank	Tiger		BigCat, Piece
	<ul style="list-style-type: none">Create Tiger piecesSet Tiger rank	<ul style="list-style-type: none">BigCatPiece	
Leopard <ul style="list-style-type: none">Create Leopard piecesSet Leopard rank	Elephant		Piece
	<ul style="list-style-type: none">Create Elephant piecesSet Elephant rank	<ul style="list-style-type: none">Piece	

CRC Cards (database)

RetrieveMatches

- | | |
|--|---|
| <ul style="list-style-type: none">Communicate with database to pull all active and past matches for a user | <ul style="list-style-type: none">Match |
|--|---|

RetrieveProfile

- | | |
|---|---|
| <ul style="list-style-type: none">Communicate with database to pull user profiles | <ul style="list-style-type: none">Profile |
|---|---|

Profile

- | | |
|--|--|
| <ul style="list-style-type: none">Hold information about each userDisplay statistics about player | |
|--|--|

Match

- | | |
|--|--|
| <ul style="list-style-type: none">Move data between JungleBoard and GamePageCommunicate with database to pull current state of gamesSet game state variables | <ul style="list-style-type: none">JungleBoardGamePage |
|--|--|



{JSON}

Traceability Link Matrix

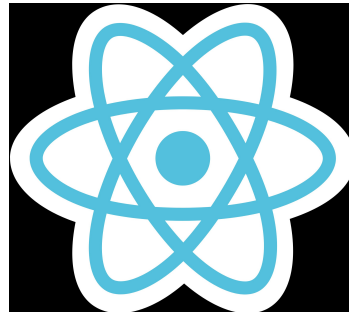
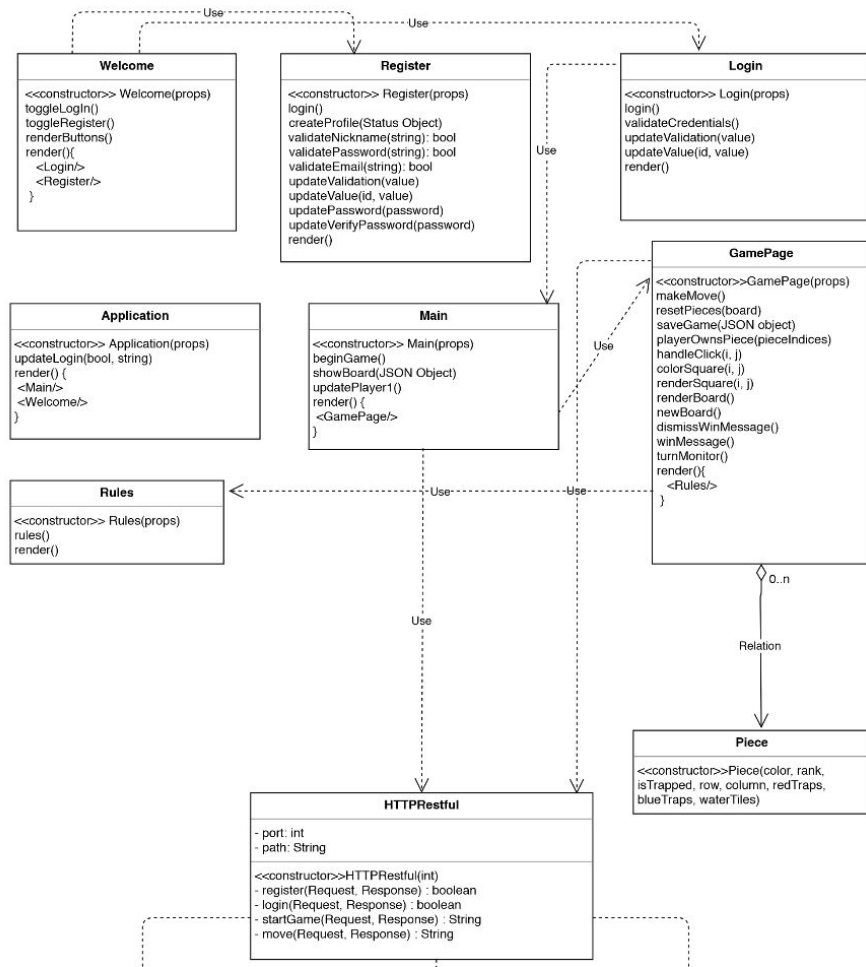
	Register	Welcome	GamePage	Rules	Login	Header	Notification	Invitation	JungleBoard
Priority 1	Register	X							
	Login		X		X				
	Play Jungle						X		X
	Invite							X	
	Accept Invite						X	X	
	Notifications							X	
	Decline Invite						X	X	
	View Current Games			X					
	View Games History			X					
	Suspend Game								
	Resume Active Game			X					
	Play Multiple Games			X					
	View Game Rules			X		X			
	Quit/Forfeit Current Game								
	View Other's Profiles								
	View Profile					X			
	Logout					X			
	Unregister								
	View Board			X					
	Chat			X			X		

***User stories implemented during the process of completing Priority 1 stories**

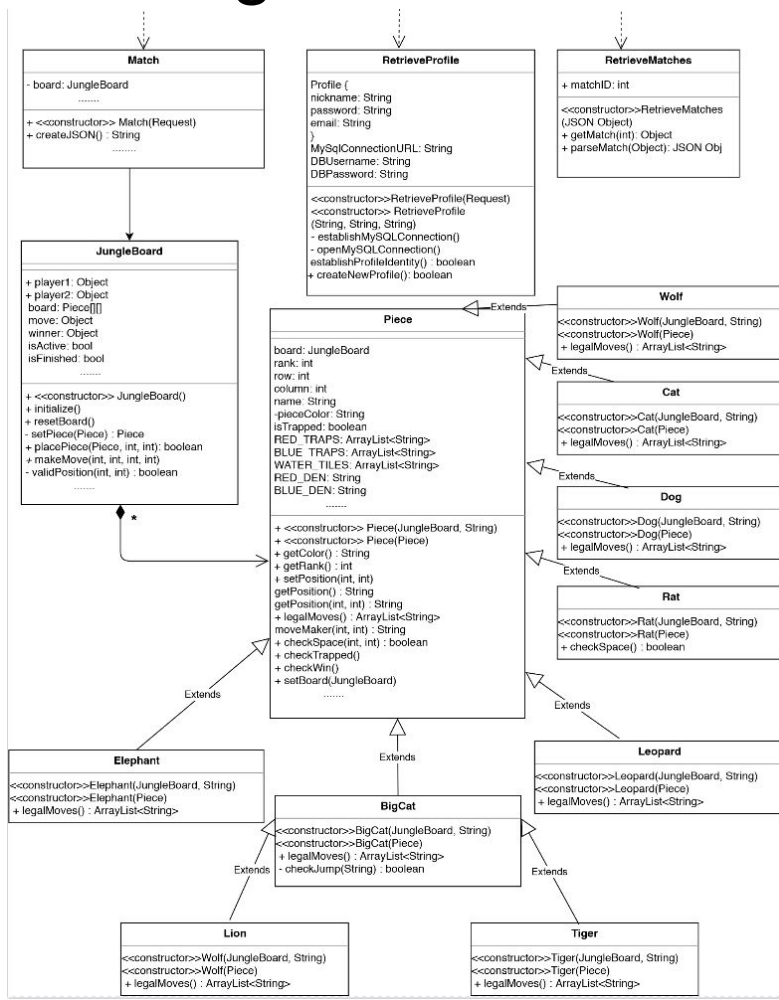
Traceability Link Matrix (cont.)

[illegible]

Class UML Diagram Client Side



Class UML Diagram Server Side



JUnit 5 Testing

- Pure JUnit 5 Methodology, no JUnit 4.
- 38 Unique Tests over 11 Test Classes.



CatTest.java	-Revamped back end to handle integer values for row and column in ord...	3 days ago
DogTest.java	-Revamped back end to handle integer values for row and column in ord...	3 days ago
ElephantTest.java	-Revamped back end to handle integer values for row and column in ord...	3 days ago
JungleBoardTest.java	-Revamped back end to handle integer values for row and column in ord...	3 days ago
LeopardTest.java	-Revamped back end to handle integer values for row and column in ord...	3 days ago
LionTest.java	-Revamped back end to handle integer values for row and column in ord...	3 days ago
PieceTest.java	-Revamped back end to handle integer values for row and column in ord...	3 days ago
RatTest.java	-Revamped back end to handle integer values for row and column in ord...	3 days ago
RetrieveProfileTest.java	Implemented and tested implementation of user registration in the dat...	8 days ago
TigerTest.java	-Revamped back end to handle integer values for row and column in ord...	3 days ago
WolfTest.java	-Revamped back end to handle integer values for row and column in ord...	3 days ago

Development Manual

Running Tests Standalone

This document explains...

- Clone the repo in IntelliJ IDEA
 - Navigate to the test directory: `src/serverTest/java/com/jungleapp/cs414/server/`
 - Right-click on the `Test` folder (or any of the bunch of `*Test.java` files)
 - click the `run 'Tests in "com.jungleapp.cs.server"'` option in the context menu
 - The test results should appear in the `run` panel (where the Terminal usually is)
- Client testing is not yet implemented. We will implement testing for client side in a future sprint using `jest`.

Database Information

Jungle uses the `java.sql.*` package to store and access data in a database. One of two databases will be used. The default, `faure`, is available for use if Jungle is run on any Linux CS department state-capital machine. Otherwise, a local database may be used.

The local database is also necessary in the case of a product demo. Due to the unavailability of a department machine during our demonstration of Jungle, the database must reside locally in the same Linux (virtual) machine as the application to ensure the demo runs smoothly.

Data Flow

The `Test` and `server` components are responsible for processing requests and returning responses to the client. The `server` component is responsible for processing requests and returning responses to the client.

Every request is listed below along with its purpose and the corresponding response. These requests are closely tied to our User Stories, so they are listed in the same order.

the JSON requests/responses

Request	Information in Response	Purpose	User Story
register	Validity of registration attempt	Allow user to create a new account	Register
login	Validity of login attempt	Allow user to login to an existing account	Login
newMatch	Newly initialized gameState	Allow user to start a new game	Play Jungle
updateMatch	gameState after move is made	Allow user take their turn	Play Jungle

Jungle Development Manual

Cloning the repo from GitHub...

..to a New Project In IntelliJ IDEA:

(IntelliJ IDEA must be in a Linux environment)

1. `New > Project from Version Control... > Git`
2. In the `URL` field, enter: `https://github.com/leejr0/cs414-f19-001-Spaghetti-Coders.git`
3. In the `Directory` field, change it from

```
/home/{user}/IdeaProjects/cs414-f19-001-Spaghetti-Coders to  
/home/{user}/IdeaProjects/cs414
```

4. Click `Clone`

Running the Application

1. Make sure the repo is cloned and up to date in IDEA.
2. In a *fresh* local terminal within IntelliJ IDEA, type `./run`
3. After compilation, the web interface should be accessible at `localhost:8090` in (most) web browsers

Information about `./run`

This command will recompile and bundle information from both the client and the server to be rendered on the web browser. After any change to the system is made, either in the client or the server, the environment must be recompiled and bundled again to see the changes implemented. All necessary files will be made automatically with the `./run` command without any extra work from the developer.

Updating the Project

To update the project, a developer should first pull any recent changes from GitHub by going to `VCS -> Git -> Pull` in the navigation at the top of the IntelliJ window. This can also be done by pressing the blue arrow in the top right corner, next to `Git`. After the project is updated, a developer should open a new branch by pressing the icon labeled `Git`: `master` in the bottom right corner of the IntelliJ window. The developer can name their branch according to the change they are making, and proceed with any changes.

Sprint Retrospective

Positives:

- Successful scrum meetings(standups and review)
- Overall system is operational
- User Stories and tasks were well documented

Negatives:

- Team communication
- Lack of pair programming
- Clear demonstration during the retrospective is needed

Jungle Demo

Image Sources:

<http://www.sitalgames.com/project/animal-chess/>

<https://dev.to/danijelajs/react-bootstrap-frameworks-review-1n13>

<https://jrebel.com/rebellabs/spark-java-is-an-amazing-java-web-framework-do-you-really-need-it/>

https://www.iconfinder.com/icons/1954836/brain_digital_electronic_processor_icon

<https://www.pinterest.com/pin/632263235162988772/?lp=true>

<https://seeklogo.net/mysql-in-eps-format-5459.html>