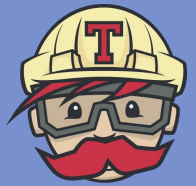
Byte Mechanics



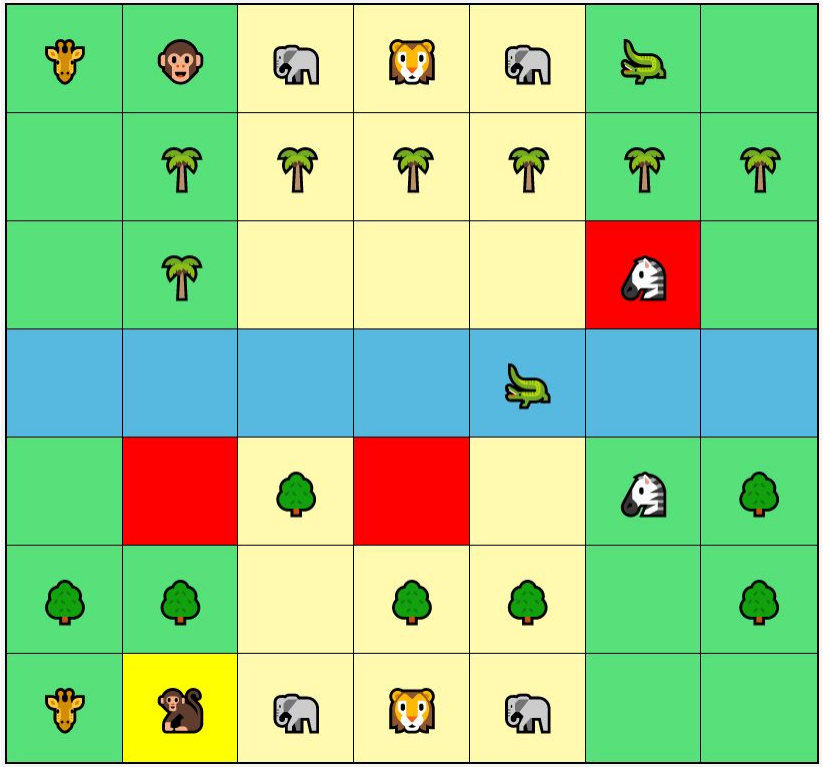
Sprint 2 Presentation

Abby Rictor, Aislinn Jeske (Scrum Master), Farzaneh Elyaderani, Marylou Nash, Zachary Klausner

Congo

High-Priority User Stories (Sprint 2)

- **Register an account**
- **Login/ logout from an account**
- **Invite a User(s) to Play Game**
- **Perform a Move on Current Game**
  - Retrieve existing game from database
  - Update board after performing valid move
- **Create a New Game (In progress)**
  - Store game and player info in database
- **Accept an Invitation (In progress)**



User Stories

High Priority Stories and Tasks (Must have)

User can register/create an account	User can login/logout to profile	User can create a new match
- User enters an email, username, and password - Check if username and email are unique - A profile will be created for user - User info added/stored in the database	- User log into profile and see information (opponent, start/end times, state) of current, finished, and abandoned games - User can see other player's profile - user can log out from an account anytime	- User choose a partner to play by sending invitation - If invitation is accepted, a new match is created - Game starts with having at least 2 players

User can invite a friend to play	User can accept the game invitation	User can make a move on current game
- User can invite a friend to play a game - User sends email/ notification for invitation	- User accept the invitation by clicking the accept button - User logs into their profile to view newly started game	- User can make a valid move on his ongoing game

CRC Cards

Game	
Responsibilities	**Collaborators**
• Create new game session • Load game session from database	GameBoard DatabaseHandler

GameBoard	
Responsibilities	**Collaborators**
• Setup new board and place pieces • Move a piece • Determine if a piece has drowned	Lion Monkey Giraffe Crocodile Pawn Zebra Elephant GamePiece

Query	
Responsibilities	**Collaborators**
• Create queries to add, update, and get information from the database	

GamePiece	
Responsibilities	**Collaborators**
• Determine if piece is in river • Validate that a move is legal • Determine if path is clear for a piece to move • Helper methods for valid moves	Lion Monkey Giraffe Crocodile Pawn Zebra Elephant GameBoard

DatabaseHandler	
Responsibilities	**Collaborators**
• Register user and store info in database • Check username and password in database so user can log in • Unregister user and remove info from database • Add new game to database • Get game from database • Save game changes to database	Action Query

Crocodile	
Responsibilities	**Collaborators**
• Validate legality of a move made by a crocodile piece based on the current state of the board	

Elephant	
Responsibilities	**Collaborators**
• Validate legality of a move made by an elephant piece based on the current state of the board	

Giraffe	
Responsibilities	**Collaborators**
• Validate legality of a move made by a giraffe piece based on the current state of the board	

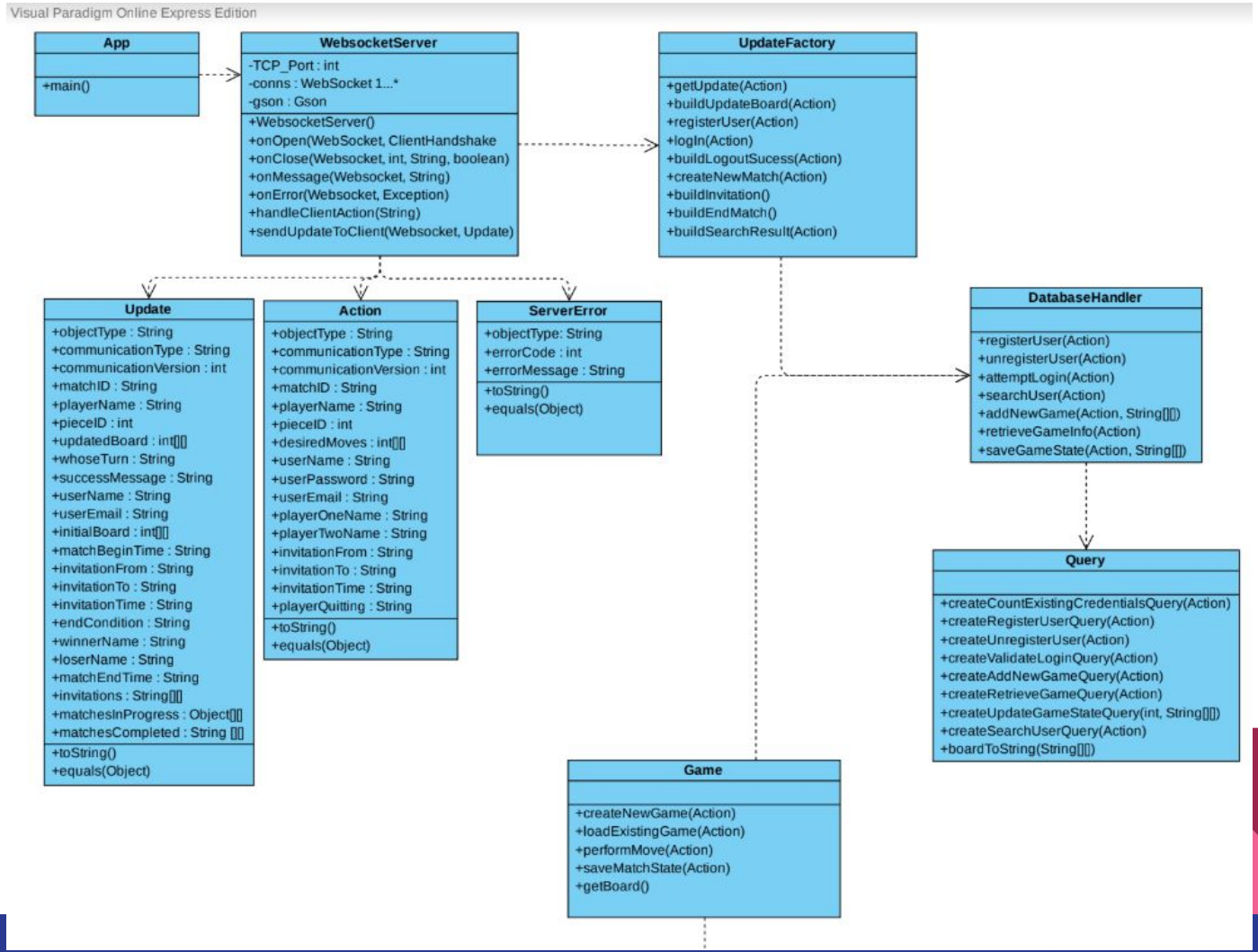
Zebra	
Responsibilities	**Collaborators**
• Validate legality of a move made by a zebra piece based on the current state of the board	

Lion	
Responsibilities	**Collaborators**
• Validate legality of a move made by a lion piece based on the current state of the board	

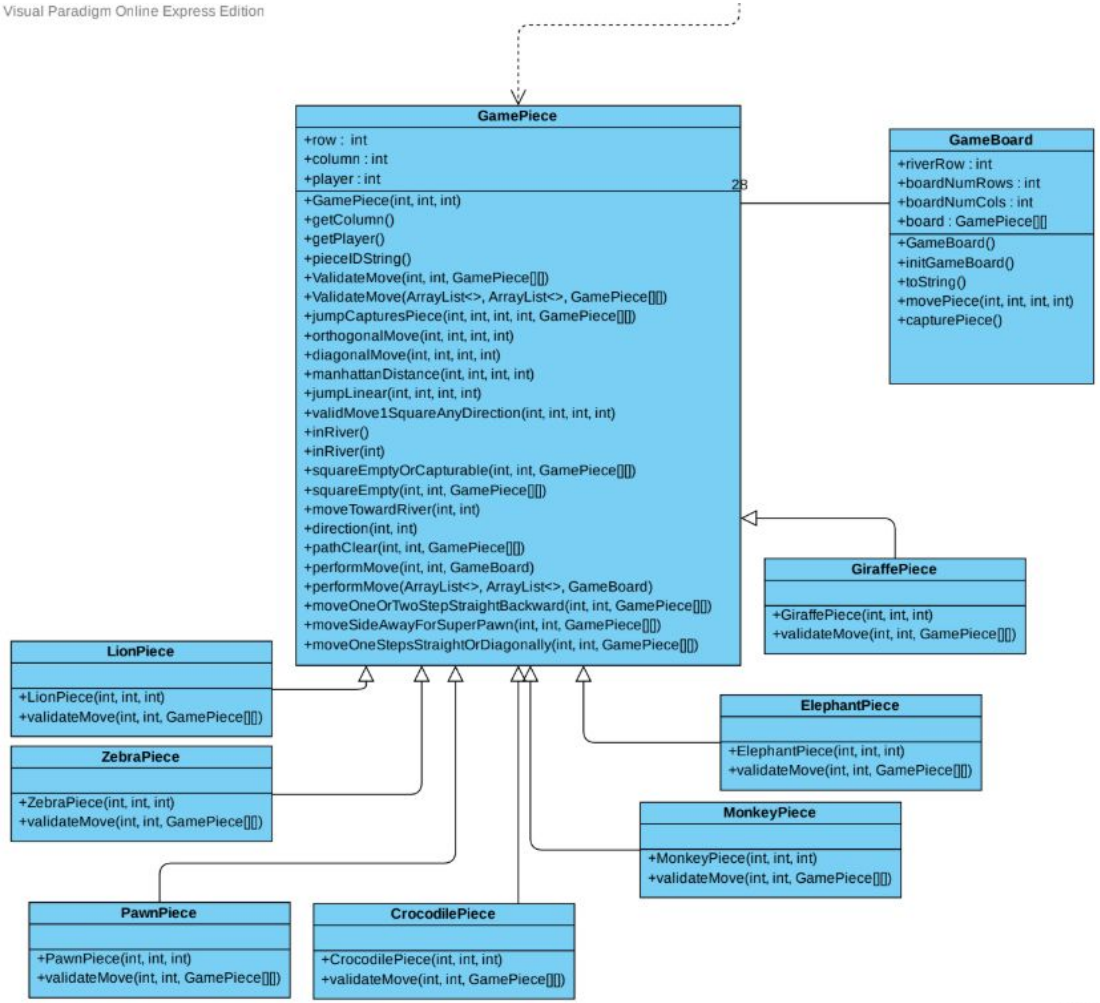
Monkey	
Responsibilities	**Collaborators**
• Validate legality of a move made by a monkey piece based on the current state of the board	

Pawn	
Responsibilities	**Collaborators**
• Validate legality of a move made by a pawn based on the current state of the board	

Class Diagram



Visual Paradigm Online Express Edition

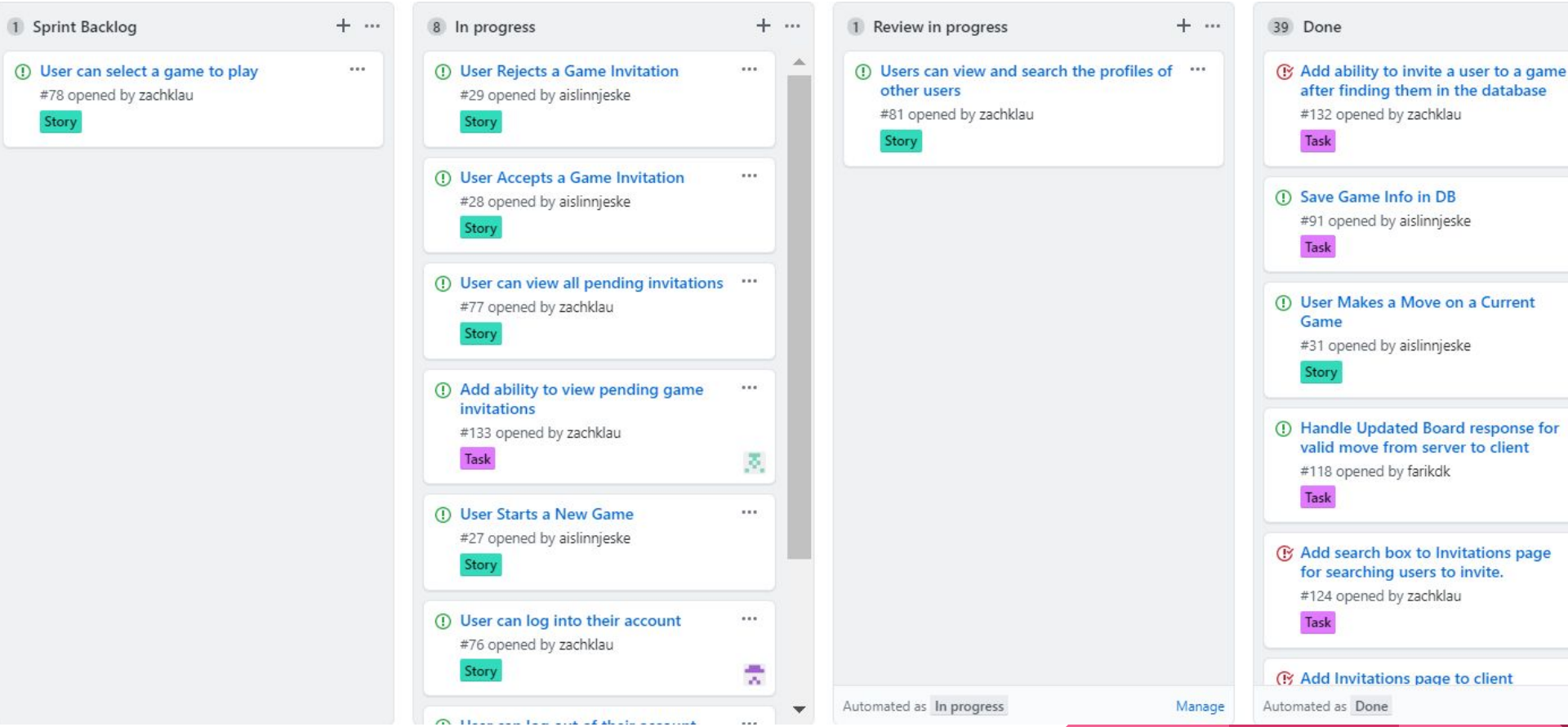


Visual Paradigm Online Express Edition

Traceability Link Matrix

A	V	W	X	Y	Z	AA	AB
	/pom.xml	Client-ServerCommFormats.md	DatabaseHandler.java	Query.java	Update.java	Action.java	UpdateFactory.java
User Story #28: User Accepts Game Invitation							
User Story #29: User Rejects Game Invitation							
User Story #30: User Abandons Game							
User Story #31: User Makes Move on Current Game							
User Story #32: User Unregisters Account							
User Story #44: Implement Drowning							
User Story #45: Implement Game Board UI							
User Story #60: User Views Games			X	X			
User Story #61: User Searches Games			X	X			
User Story #69: Create and Organize a Tournament			X				
User Story #70: Distribute Awards for Tournament Results			X				
User Story #71: Users can Message Other Users							
User Story #72: Users can Play Against an AI Oppoment					X		X
User Story #73: Keep Track of Player's Ranking and Badges			X				
User Story #74: User Can Play Multiple Games at Same Tim							
User Story #75: Keep Track of Player's Game History			X	X			
User Story #76: User Can Log into their Account							
User Story #77: User Can View All Pending Invitations				X			X
User Story #78: User Can Select a Game to Play			X	X	X		X
User Story #79: User Can Close (& Save) Game & Select Ar			X	X	X		X
User Story #80: User Can Log Out of their Account							
User Story #81: User Can View & Search Profile of other Pla			X	X			X

Kanban Board



Review and Retrospective

- **What work did we commit to being delivered?**
  - Storing a game in database
  - Search for players to invite, Send game invitations, Accept/Decline Invitations, View invitations
  - Making a move on the board
  - User login/logout, User registration, User unregisters
- **What work did we complete?**
  - Storing a game in database
  - Searching for players to invite, Sending game invitations
  - Making a move on the board
  - User login/logout, User registration
- **What did we do well?**
  - Refactoring, working on other parts of the code base, communication, changing implementation according to design needs, flexibility, understanding full project stack
- **What should be focus more on to improve?**
  - Tackle bigger more extensive changes early in sprint to allow time to adjust
  - Smaller and incremental changes, merging earlier and more often
  - Anticipating structural needs for future features

Questions/Demo