

Team 9

Name and ID:

Hua Wang, 40011373

Mengyi Li, 26793827

Tianyi Hu, 27595638

Category:

- Requirements
 1. Functional Requirements
 2. Non-Functional Requirements
 3. User Stories
- Use Cases
 1. Use Case Diagram
 2. Sequence Diagram
 3. Use Case
- Design
- Architecture
 1. Model view controller
 2. Architectural Views
- GUI Decisions
- Testing Plans
 1. Unit Test
 2. Integration Test
 3. Acceptance Test
 4. Regression Test
 5. System Level Test
- System Behaviour Analysis
- Scoping/Cost Considerations
- Installation and Commissioning Instructions

Function Requirements

| | | Functional Requirements |
|---|-------------------------|---|
| | Requirements | Description |
| 1 | Game Starts | We need to implement register, log in, and log out to make sure the game will be start successfully. Then we need to choosing deck. |
| 2 | Decks Parse | we need to implement the deck parse |
| 3 | Cards Dealt | It is a function, which will handle shuffle and dealt cards to both of the players. |
| 4 | Mulligan | If either player have no Basic Pokemon in their opening hand, that player must take a mulligan. |
| 5 | Play Initial Pokemon | To start the game, both of the players need to choose a pokemon to be the active pokemon. |
| 6 | Bench Initial Pokemon | We can choose to put the rest of the pokemon on the bench or not. It does not matter how many pokemons player put on the bench. |
| 7 | Draw Card at Turn Start | Every turn the player can choose one card from their card pile to make sure the game can be continue. |

| | | |
|----|-----------------------|--|
| 8 | Play Items | When the game goes, we can using trainer cards to help us to win the game by following the rules of the cards. |
| 9 | Bench Pokemon | We can decide to retreat pokemon to the bench or put the pokemon from your hand to the bench. |
| 10 | Evolve Pokemon | If we have higher stage of our Pokemon and also have a basic stage of the pokemon, then we can choose to evolve a Pokemon. |
| 11 | Place Energy | Every ability or attack need to meet the need of the number of energy so that we need to attach some energy on the pokemon card. |
| 12 | End Turn | To end turn, there should be a loser or a winner of a turn |
| 13 | Use Pokemon | To start new turn and if the player lost, we need to choose a new active Pokemon and choose it from a bench |
| 14 | AI Plays | In this game app, we only play with AI player and the AI player should give feedback like a real player |
| 15 | Retreat Pokemon | To protect the active Pokemon not been knock out, we can retreat the active pokemon with a bench Pokemon. However, the retreat Pokemon will lose all the energy. |
| 16 | Knock Out Pokemon | If the damage meet the Health Point of one pokemon, this pokemon will be knock out |
| 17 | Collect Prize Card | If a player knock out a pokemon from his opponent, he can choose one of the prize card from the pile. |
| 18 | Win | 1.A player grab all his own prize cards. 2.The player's opponent do not have pokemon 3.The player's opponent do not have any other card. |
| 19 | Lose | 1.The player's opponent grab all his own prize cards. 2.The player do not have pokemon 3.The player do not have any other card. |
| 20 | Check Both Hand Sizes | We should be able to see how many cards we have in hand |
| 21 | Check Both Deck Sizes | be able to check a deck's size |

| | | |
|----|---------------------------------|--|
| 22 | Check Both Discard Sizes | be able to check the size of the discard pile |
| 23 | Look At Discards | Some of the trainer cards can look at a discard pile so that we need to implement this function. |
| 24 | Check Energy on Active Pokemon | Be able to check how many energy the active Pokemon have to decide we can use its ability or not. |
| 25 | Check Energy on Benched Pokemon | Be able to check how many energy the active Pokemon have to decide we can use its ability or not. |
| 26 | Look at Pokemon Ability | We should be able to know what ability the pokemon have. |
| 27 | Look at Ability of Card in Hand | We also should be able to see the ability of a card in hand to know if it's better than active Pokemon, in case if we want to retreat, |
| 28 | See Current Pokemon Health | We should be able to see Pokemon's current health point so that we can decide we retreat the pokemon or not. |
| 29 | See Max Pokemon Health | We should be able to see Pokemon's max health point so that we can decide strategy properly. |

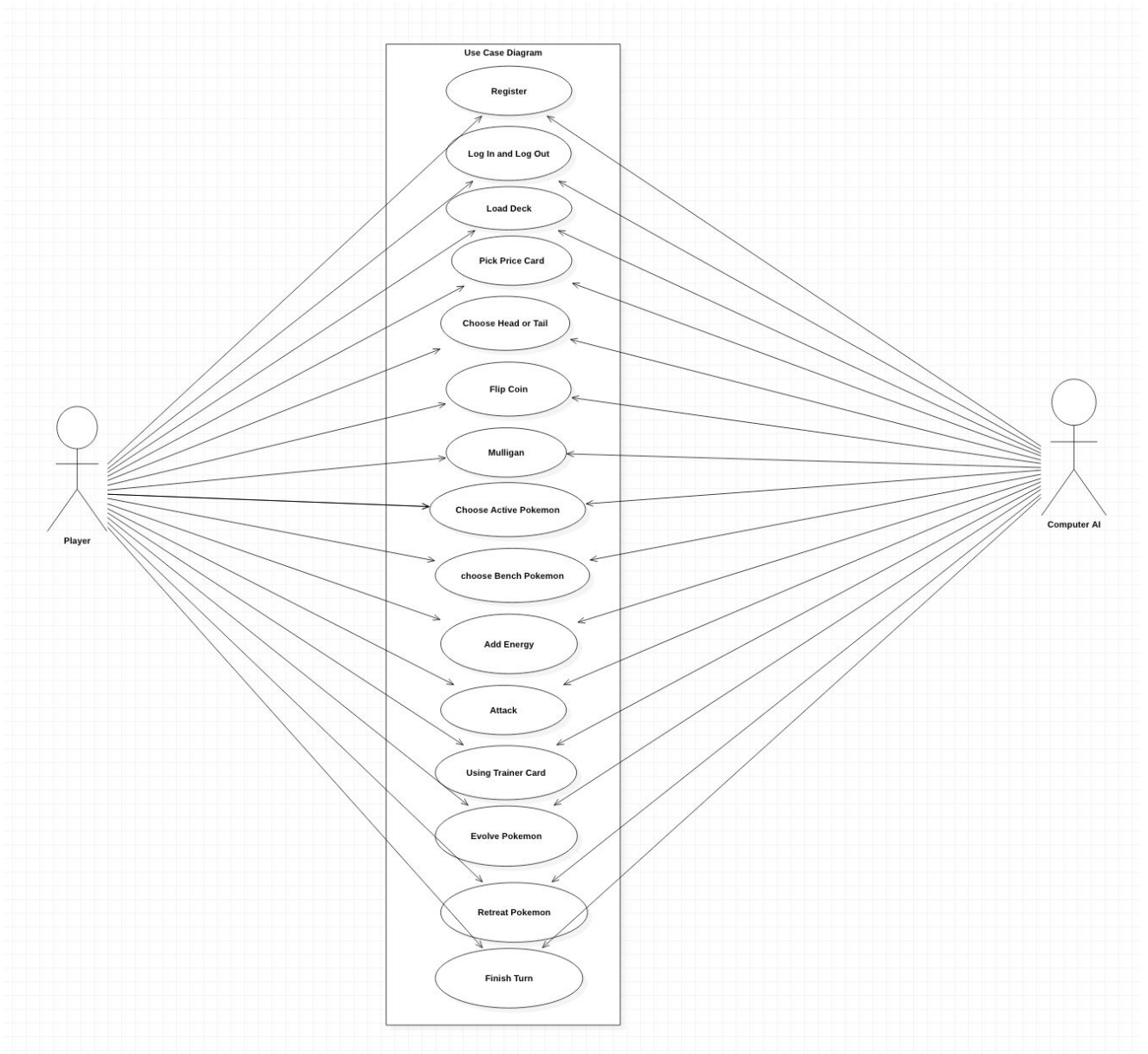
Non-Functional Requirements

| | Requirement | Description |
|---|---------------|--|
| 1 | Performance | Should be fast and dynamic |
| 2 | Storage | Github |
| 3 | Environmental | Java, React JS, IntelliJ IDEA, Spring boot |

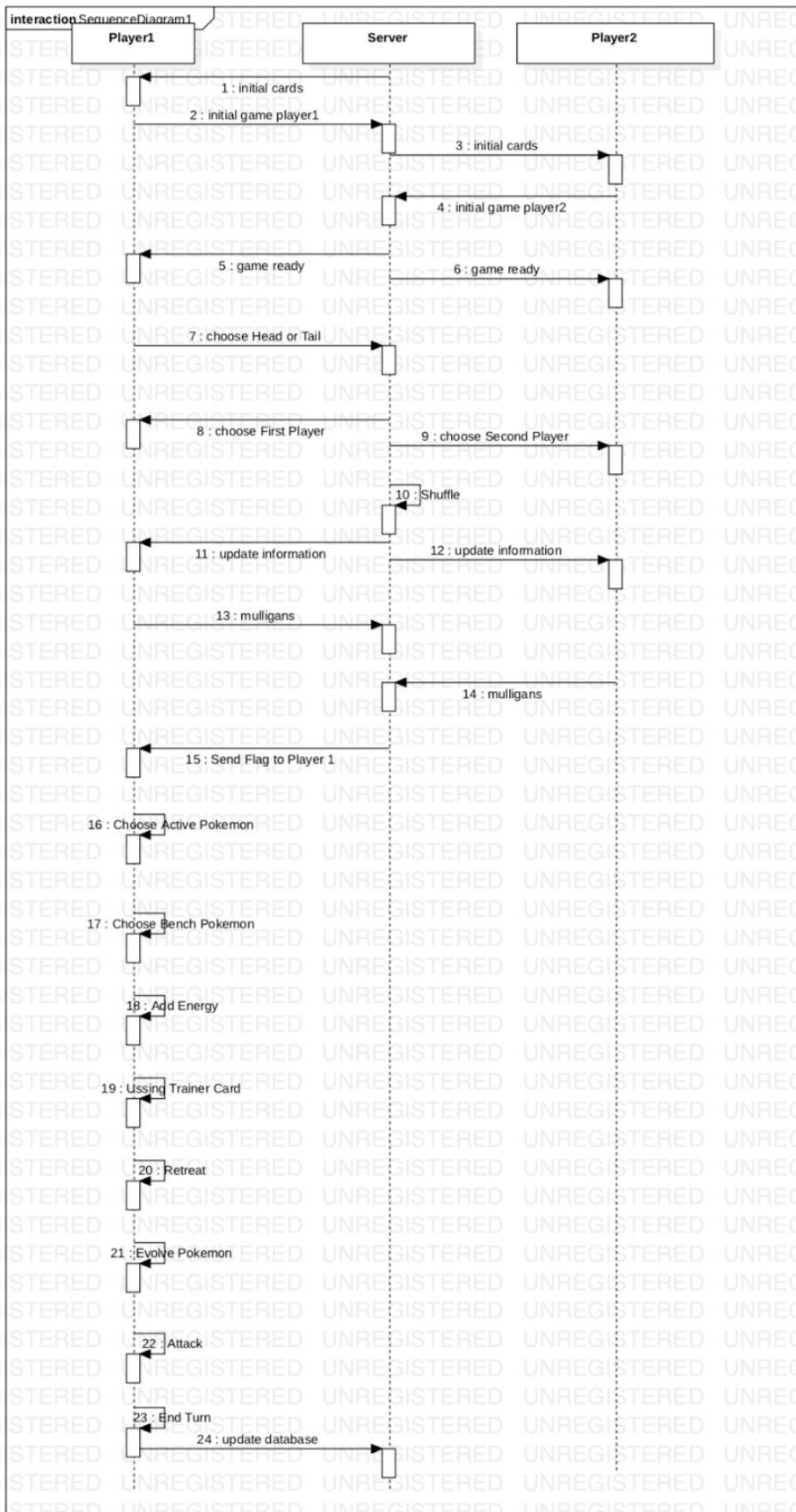
User story

| Task | Description |
|------------------------------|---|
| 1 .Read the rule of the game | Understand how the rule |
| 2. Cardså | Create card object 1.Pokemon 2. Energy 3. Trainer |
| 3.Player | Create the Player role |
| 4.Board | Create the game board |
| 5.Game Starts | Register and log in |
| 6.Decks Parse | Implement the deck parse |
| 7. Shuffle | every player should have 7 cards when game stard |
| 8. Mulligans | After Shuffle, if there is no Pokemon cards in your hands , shuffle the cards again |
| 9.Play Initial pokemon | Choose a Pokemon as the active Pokemon |
| 10. Bench Initial pokemon | Choose rest of cards on the bench |
| 11.Draw Card at turn Start | Every turn the player need to choose one action |
| 12. Play Items | Using trainer cards help player |
| 13. Bench pokemon | We can decide to retreat Pokemon in the bench |
| 14. Evolve Energy | If we have higher stage Pokemon, we can evolve it |

| | |
|-------------------------------------|--|
| 15. Place Energy | Help Pokemon attack or retreat or use ability |
| 16. End Turn | |
| 17. Use Pokemon | |
| 18. AI Plays | Develop AI algorithm player agist player |
| 19. Retreat Pokemon | Protect the active Pokemon not been knock out |
| 20. Knock Out Pokemon | If the damage meet the Health Point of Pokemon, this Pokemon will be knock out |
| 21. Collect Prize Card | If a player knock out a Pokemon, he can choose one of the prize card |
| 22. Win | 1. A player grab all his own prize cards. 2. The player's opponent do not have Pokemon 3. The player's opponent do not have any other card |
| 23. Lose | Similar as Win |
| 24. Check Both Hand Sizes | Confirm the amount of cards in hand |
| 25. Check Both Deck Sizes | Check the size of deck |
| 26. Check Both Discard Sizes | Confirm the amount of cards in discard pile |
| 27. Look At Discards | |
| 28. Check Energy On Active Pokemon | Check how many energy the active Pokemon's ability |
| 29. Check Energy On Benched Pokemon | Check how many energy the Pokemon in bench |
| 30. Look at Pokemon Ability | Know what ability of Pokemon |
| 31. Look at Ability of Card in Hand | |
| 32. See Current Pokemon Health | Check hp |
| 33. See Max Pokemon Health | Check max hp to decide strategy |
| 34. AI design | Find a AI search algorithm |
| 35. UI | |
| 36. Log in | Log in the server |
| 37. Log out | Log out the server |
| 38. User Management | Make sure |



Sequence Diagram



Use Case

| Name | Description |
|--------------------------------|---|
| Scope | Register |
| Level | Register a name and password from webpage |
| Primary Actor | Player |
| Stakeholder and Interests List | none |
| Preconditions | none |
| Postconditions | none |
| Main Success scenarios | none |
| Extensions | none |

| Name | Description |
|--------------------------------|--|
| Scope | Log In and Log Out |
| Level | use password and user name to log in and log out the webpage |
| Primary Actor | Player |
| Stakeholder and Interests List | none |
| Preconditions | 1.The user must register a user name and password successfully. 2.The user must input the right user name and the password. |
| Postconditions | 1.Player can prepare to play game when they login |
| Main Success scenarios | none |
| Extensions | none |

| Name | Description |
|--------------------------------|---|
| Scope | Load Deck |
| Level | Load Deck and other information from database |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | There should be at least two Decks in the database |
| Postconditions | 1.Both players Decks should be in the table 2.Both players must choose their own deck. |
| Main Success scenarios | none |
| Extensions | none |

| Name | Description |
|--------------------------------|--|
| Scope | Pick Prize Card |
| Level | Both of the players need to pick six cards from their deck to be the prize card. |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | Choosing 6 cards to be your Prize card |
| Postconditions | 1.Every time one of your opponent's Pokemon is Knocked Out, you take one of your Prize cards and put it in your hand. 2.If Player choose all of the six card, this player will be win and game is over. |
| Main Success scenarios | none |
| Extensions | none |

| Name | Description |
|--------------------------------|--|
| Scope | Choose Head or Tail |
| Level | Choosing Head or Tail from the coin and the computer will flip the coin automatically to decide who will be the first one to play. |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | none. |
| Postconditions | The winner decides which player goes first. |
| Main Success scenarios | none |
| Extensions | none |

| Name | Description |
|--------------------------------|--|
| Scope | Flip coin |
| Level | There are some special conditions need to be check and decide before game continue. |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | <ul style="list-style-type: none"> • Asleep condition: If a Pokemon is Asleep, it cannot attack or retreat. • Burned condition:A Burned Pokemon takes damage between turns, but the condition might heal on its own. When a Pokemon is Burned, put a Burn marker on it. Between turns, put 2 damage counters on your Burned Pokemon. • Confused condition:If Pokemon is Confused, player must flip a coin before attacking with it. |
| Postconditions | <ul style="list-style-type: none"> • Asleep condition: flip a coin, if you flip heads, the Pokemon wakes up, but if you flip tails, it stays Asleep. • Burned condition:flip a coin. If heads, remove the Special Condition. • Confused condition:flip a coin. If heads, the attack works normally. If tails, the attack doesn't happen, and you put 3 damage counters on your confused Pokemon. |
| Main Success scenarios | none |
| Extensions | <ul style="list-style-type: none"> • Trainer card condition: some of special trainer card have some different kinds of rule and it also need to flip a coin to decide it is positive effect or not. |

| Name | Description |
|--------------------------------|--|
| Scope | Mulligan |
| Level | If either player have no Basic Pokemon in their opening hand, that player must take a mulligan. |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | <p>1.If both players have no Basic Pokemon in their opening hands.</p> <p>2.If only one player has no Basic Pokemon in their opening hand.</p> |
| Postconditions | <p>1.Both player reveal their cards, then just start over as normal.</p> <p>2-a: That player announces that they have a mulligan, then waits until the other player has finished setting up to play.</p> <p>2-b:Then the player with no Basic Pokemon reveal their cards and shuffles it back into the deck. The player keep doing this until they get an basic Pokemon.</p> |
| Main Success scenarios | none |
| Extensions | none |

| Name | Description |
|--------------------------------|--|
| Scope | Choose active pokemon |
| Level | Choose one of the basic pokemon to be the active pokemon. |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | <p>1. There should be an Active Pokemon in your hand if you start the game.</p> <p>2. If your Pokemon is defeated, there should be a Pokemon on a bench.</p> |
| Postconditions | An active Pokemon should be chosen from the hand or a bench. |
| Main Success scenarios | none |
| Extensions | none |

| Name | Description |
|--------------------------------|---|
| Scope | Choose bench pokemon |
| Level | Choose one of the pokemon to be the bench pokemon. |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | <p>1. There should be an Pokemon in your hand if you decide to choose a pokemon on the bench.</p> <p>2. There should be one Pokemon in the active pokemon place so that you can retreat it back to the bench.</p> |
| Postconditions | If your Pokemon is defeated, there should be a Pokemon on a bench |
| Main Success scenarios | none |
| Extensions | none |

| Name | Description |
|--------------------------------|---|
| Scope | Add Energy |
| Level | It adds energy to an active Pokemon or to a bench Pokemon. |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | There should be an energy card to choose from your hand and a Pokemon card. |
| Postconditions | the energy card is no more in your hand, and the energy will be attached to a Pokemon Card. |
| Main Success scenarios | none |
| Extensions | none |

| Name | Description |
|--------------------------------|---|
| Scope | Attack |
| Level | Using your active Pokemon to attack your opponent and Attack the opposite player following the rules of the ability of the active Pokemon |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | An Active Pokemon should have enough energy to attack. |
| Postconditions | 1.Pokemon cannot attack if it is knocked out 2.The opponent pokemon should be damaged with the attack ability the player using for. |

| | |
|------------------------|------|
| Main Success scenarios | none |
| Extensions | none |

| Name | Description |
|--------------------------------|---|
| Scope | Using Trainer Card |
| Level | The player will follow the rules on specified on a trainer card |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | Multiple Trainer cards can be played in one turn |
| Postconditions | the Trainer card won't be in your hand, and it will be attached to the Pokemon card |
| Main Success scenarios | none |
| Extensions | none |

| Name | Description |
|--------------------------------|---|
| Scope | Evolve Pokemon |
| Level | Evolving a Pokemon from basic stage to stage 1 |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | 1. There should be an active Pokemon or at least a basic Pokemon on the bench that we decide to evolve. 2. There should be a Pokemon can be evolve. 3. The player cannot evolved the pokemon on the first turn. |
| Postconditions | 1. A Pokemon will evolve and have more HP and more higher damage abilities. 2. One turn can only evolve one pokemon. |
| Main Success scenarios | none |
| Extensions | none |

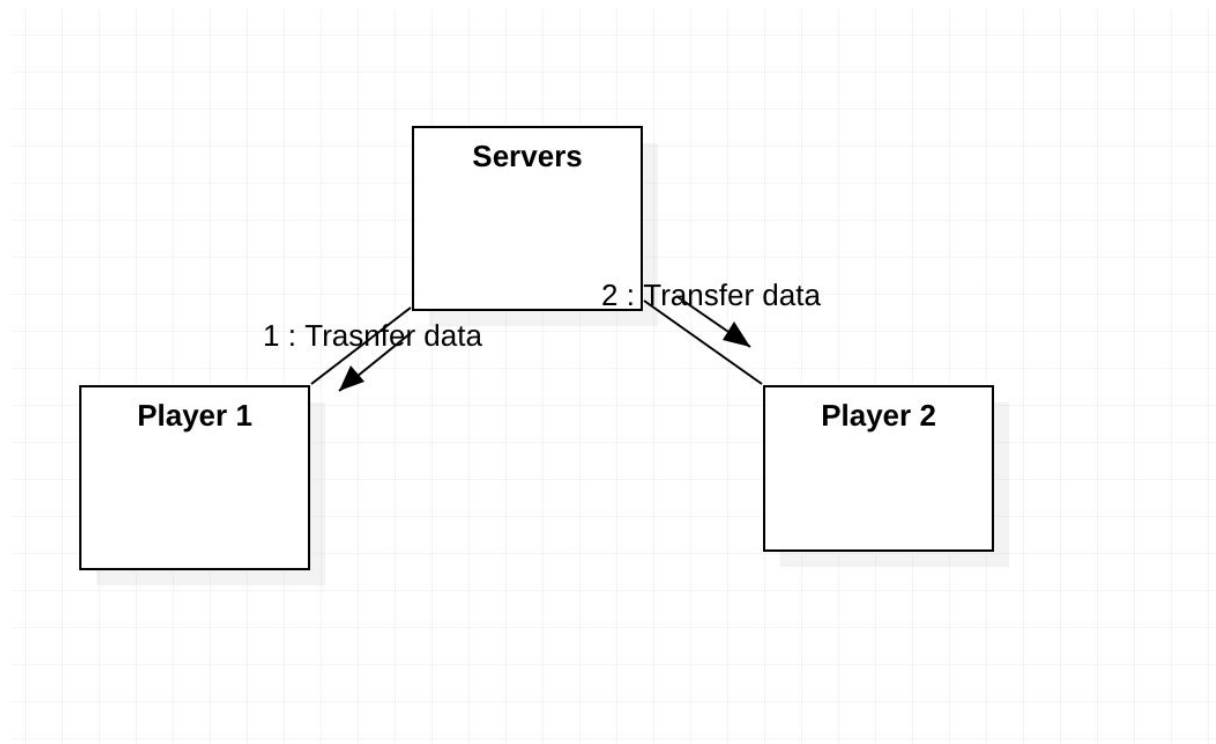
| Name | Description |
|--------------------------------|--|
| Scope | Retreat Pokemon |
| Level | Switch active Pokemon with bench Pokemon |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | <ul style="list-style-type: none"> • There should be an active Pokemon • There should be a Pokemon on a bench • A Pokemon cannot be evolved on the first turn of the player • A Pokemon cannot get retreated if a Pokemon is asleep or paralyzed |
| Postconditions | <ul style="list-style-type: none"> • Active Pokemon will end up on a bench • Bench Pokemon becomes Active Pokemon |
| Main Success scenarios | none |
| Extensions | none |

| Name | Description |
|--------------------------------|--|
| Scope | Finish Turn |
| Level | Switch Player after you apply attack. Or winning the game if the opponent have no pokemon. |
| Primary Actor | Player, Computer AI Player |
| Stakeholder and Interests List | none |
| Preconditions | Appling Attack or pass the attack option. |
| Postconditions | none. |
| Main Success scenarios | none |
| Extensions | none |

Design

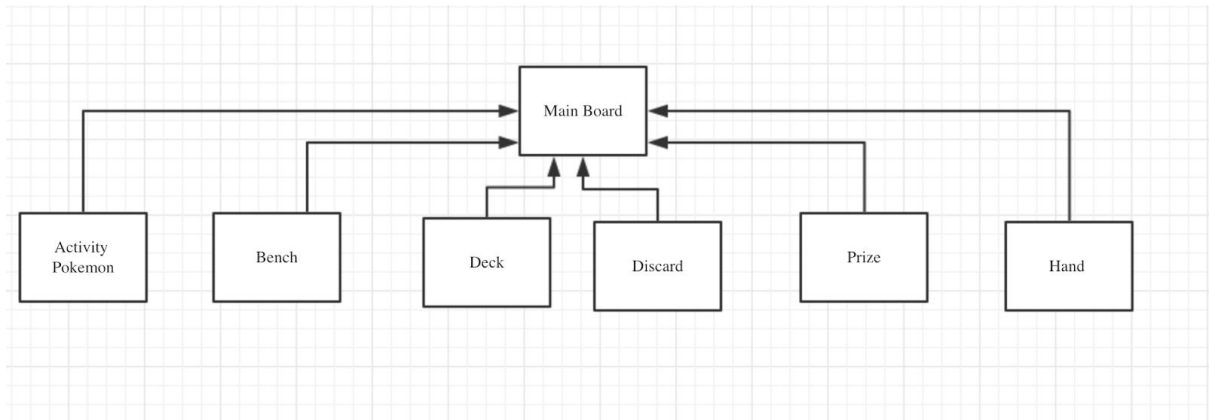
1. Server-side and Client -side
2. Game logic
 - a. Rule of game
 - b. Win or lose condition
3. Card movement
 - a. Shuffle
 - b. active Pokemon
 - c. attach energy
 - d. attack
 - e. use ability
4. Register System
 - a. User Log in and Log out
 - b. Match Player
5. Database for data store
 - a. Store data

Architeture



The game consist with 2 part , Client-side and Server-side. Server -side contain Register system and database. Client-side contain the main part of the game, depending on the user's choice of actions, the Server will use the function of client side.

GUI decisions



Main Board consist with 6 part.

1. Activity Pokemon
2. Bench Pokemon
3. Deck
4. Discard
5. Prize card
6. Hand

Unit Testing

Unit Test is the most basic and the most important test we need to do at the beginning.

How to Test example:

For example, if you want to test Register function, you just need to input the website and click register directly.

User Management:

Register

Login

Logout

- DeckManagement:

Upload

Delate

Set

Game Management:

Match Players

Start a New Game

Receive Game Data

Send New Data

Record game data

CurrentPlayerControl: the server cannot receive Players information in the same time.

Integration Test

In integration testing, individual modules are combined and tested as a group. There are many different types of integration testing, our strategy is to use the bottom-up approach, where the lowest level components are tested first, then used to facilitate the testing of higher level components.

Move Card:

- Draw Card
- Shuffle
- Attach
- Evolve
- Choose Card From List

After Move Card:

- Change

Change:

- Ability
- Attribute

After Change:

- Evaluate the winner
- Discard

Quit Game

Acceptance Testing:

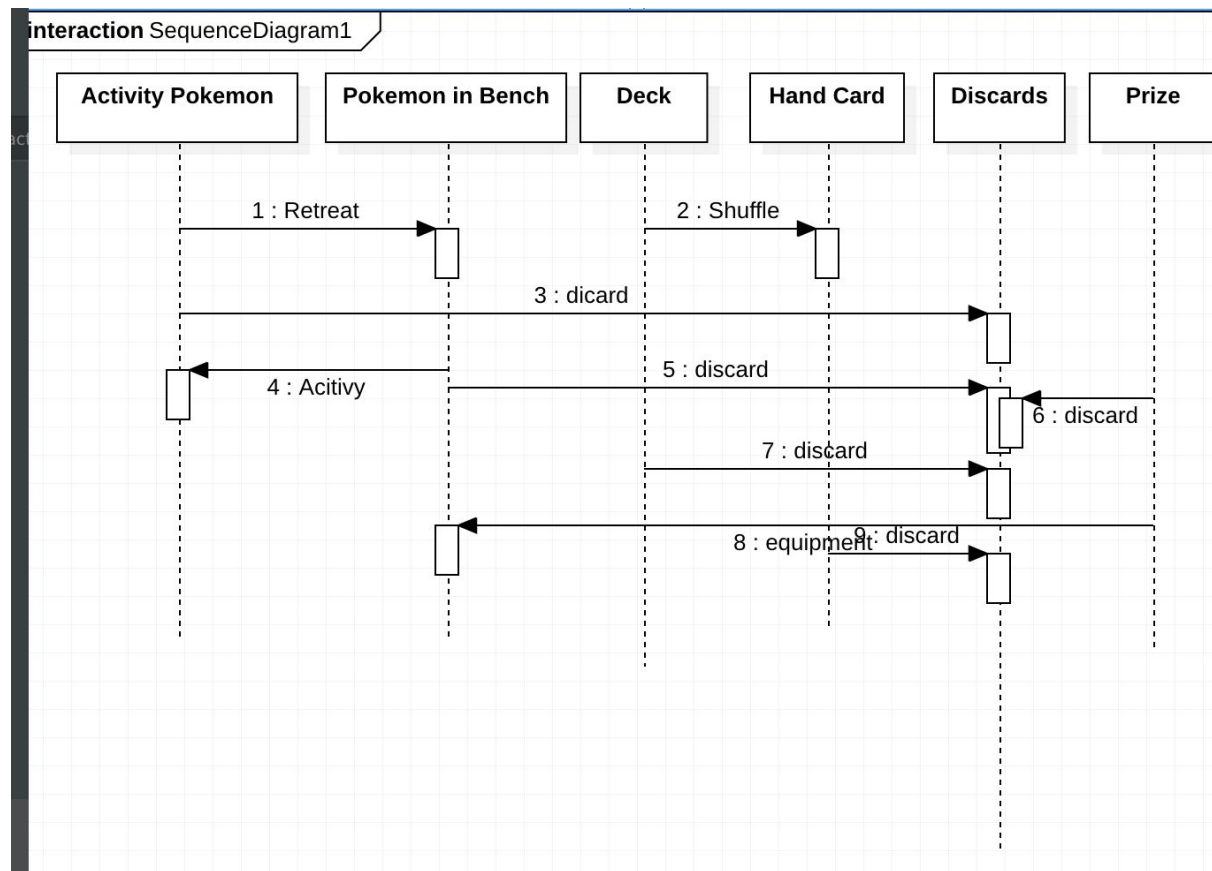
Regression Test:

- Find a bug, write a test that catches it.
 - That bug is fixed when regression tests passes□.
- That bug should never come back.

System Level Test:

System Behaviour Analysis

Movement of cards



Movement of cards show the basic rule of the game

Scoping/Cost Consideration

| Task | Description | Status | Difficulty |
|------------------------------|---|-----------------|------------|
| 1 .Read the rule of the game | Understand how the rule | Done | Easy |
| 2. Cardsâ | Create card object 1.Pokemon 2. Energy 3. Trainer | Implemented | Easy |
| 3.Player | Create the Player role | Implemented | Easy |
| 4.Board | Create the game board | Not Implemented | Medium |
| 5.Game Starts | Register and log in | Implemented | Easy |
| 6.Decks Parse | Implement the deck parse | Not Implemented | Medium |
| 7. Shuffle | every player should have 7 cards when game stard | Not Implemented | Easy |
| 8. Mulligans | After Shuffle, if there is no Pokemon cards in your hands , shuffle the cards again | Not Implemented | Easy |
| 9.Play Initial pokemon | Choose a Pokemon as the active Pokemon | Not Implemented | Easy |
| 10. Bench Initial pokemon | Choose rest of cards on the bench | Not Implemented | Easy |
| 11.Draw Card at turn Start | Every turn the player need to choose one action | Not Implemented | Medium |
| 12. Play Items | Using trainer cards help player | Not Implemented | Easy |

| | | | |
|------------------------------|---|-----------------|--------|
| 13. Bench pokemon | We can decide to retreat Pokemon in the bench | Not Implemented | Medium |
| 14. Evolve Energy | If we have higher stage Pokemon, we can evolve it | Not Implemented | Easy |
| 15. Place Energy | Help Pokemon attack or retreat or use ability | Not Implemented | Easy |
| 16. End Turn | | Not Implemented | Easy |
| 17. Use Pokemon | | Not Implemented | Easy |
| 18. AI Plays | Develop AI algorithm player agist player | Not Implemented | Medium |
| 19. Retreat Pokemon | Protect the active Pokemon not been knock out | Not Implemented | Easy |
| 20. Knock Out Pokemon | If the damage meet the Health Point of Pokemon, this Pokemon will be knock out | Not Implemented | Easy |
| 21. Collect Prize Card | If a player knock out a Pokemon, he can choose one of the prize card | Not Implemented | Medium |
| 22. Win | 1.A player grab all his own prize cards. 2. The player's opponent do not have Pokemon 3.The player's opponent do not have any other card | Not Implemented | Easy |
| 23. Lose | Similar as WIn | Not Implemented | Easy |
| 24. Check Both Hand Sizes | Confirm the amount of cards in hand | Not Implemented | Medium |
| 25. Check Both Deck Sizes | Check the size of deck | Not Implemented | Medium |

| | | | |
|-------------------------------------|--|-----------------|--------|
| 26. Check Both Discard Sizes | Confirm the amount of cards in discard pile | Not Implemented | Medium |
| 27. Look At Discards | | Not Implemented | Easy |
| 28. Check Energy On Active Pokemon | Check how many energy the active Pokemon's ability | Not Implemented | Easy |
| 29. Check Energy On Benched Pokemon | Check how many energy the Pokemon in bench | Not Implemented | Easy |
| 30. Look at Pokemon Ability | Know what ability of Pokemon | Not Implemented | Easy |
| 31. Look at Ability of Card in Hand | | Not Implemented | Easy |
| 32. See Current Pokemon Health | Check hp | Not Implemented | Easy |
| 33. See Max Pokemon Health | Check max hp to decide strategy | Not Implemented | Easy |
| 34. AI design | Find a AI search algorithm | Not Implemented | Medium |
| 35. UI | | Not Implemented | Medium |
| 36. Log in | Log in the server | Implemented | Easy |
| 37. Log out | Log out the server | Implemented | Easy |
| 38. User Management | Make sure | Implemented | Medium |
| 39. Store Data | Store data for post-mortem | Not Implemented | Medium |
| 40. Improve UI | | Not Implemented | Hard |
| 41. Improve AI | Find a more efficient algorithm | Not Implemented | Hard |
| 42. Register | Create a account for Player | Implemented | Medium |
| 43. match player | Finding AI player | Not Implemented | Medium |
| 44. Synchronization data | Store the data into database | Not Implemented | Hard |
| 45. Operation command | | Not Implemented | Medium |

[illegible]

Installation and commisioning

1. Run on Windows and OS
2. Open Compiler
3. Open Pokemon pakage
4. Run the `mvn spring-boot:run`