SOFTWARE ENGINEERING PROCESS (SOEN 6011)

Software Requirements Specification, System Design Specification, and Test Document

PokémonGoBack Game

Submitted to:

Dr. Stuart.Thiel

Prepared by:

Student Name	Student ID
Arezou Mofidi	27664443
Arvinder Singh	40021973
Amanjot Kaur	40036314
Chetan Sharma	40017009
Joy Dey	27880197
Karun Sharma	40016964
Kourosh Aziznejad	40014215
Mahdi Etminan	27116543
Malek Atwiz	26984371
Mandeep Ghuman	40011904
Umair Ashraf	27748132

Date: 2017/06/22

Version 1.0

Roles and Responsibilities

NAME	ROLE	DESCRIPTION
Arezou Mofidi	Documenter	
Arvinder Singh	Tester	
Amanjot Kaur	Documenter	
Chetan Sharma	Tester	
Joy Dey	Tester	
Karun Sharma	Tester	
Kourosh Aziznejad	Developer	
Mahdi Etminan	Developer	
Malek Atwiz	Developer	
Mandeep Ghuman	Documenter	
Umair Ashraf	Developer	

Table of Content

SECTION II

List of Tables

List of Figure

SECTION I: System Requirement Specification

1

Introduction

Goal of this project is to develop an application ‘PokémonGoBack’ game. This application is all about card game, in which various advanced features such as “GUI” and “one-player mode” are implemented. This application will be developed for stand-alone PC.

While creating this application we will keep in mind as this game is card game, so each card should be valid i.e. it should have name, type, possibly a category and one or more abilities. Each card should be either a Pokémon card, Energy card or a Trainer card. Further, each Pokémon can have any of the category as Basic, Stage-one, psychic and fighting. Energy card can be colorless, water, lightning and Trainer card can be Supporter, Item, and Stadium. For developing this application we are provided with two Decks as deck1.txt and deck2.txt. To be able to interact with we need to have 28 unique cards in deck 1 and 24 cards in deck 2.

This game switches between Player and his Opponent. Player plays as Pokémon and Attack on its Opponent Pokémon by performing several actions such as Evolving their Pokémon, playing Trainer card and Energy card or using Pokémon Abilities. Players begin the game by drawing seven cards from their shuffled Deck, placing six cards on side as Prize cards and one Basic Pokémon in the field as Active Pokémon. Which player will go first is decided by flipping the coin. Player cannot Attack in his first turn. Player face up one Prize card if he successfully knocked out Opponent Pokémon on Attack. Player who draws all the Prize cards, Knocked out all Opponent’s Pokémon announced as winner.

2

Requirement Analysis

In this section we describe all functional specifications that we used to develop PokémonGoBack project in first submission. Figure 1 shows a high level graphical view of the major functionality of the product (for first delivery). It shows the main use case of the system. The following sections will describe each use cases that shows in figure 1 in three ways. First, the description part which describes its main success scenario and extensions (alternatives). Next, there are System Sequence Diagram (SSD) and Activity Diagram for each use case in order to have a better view about the functionality of the system.

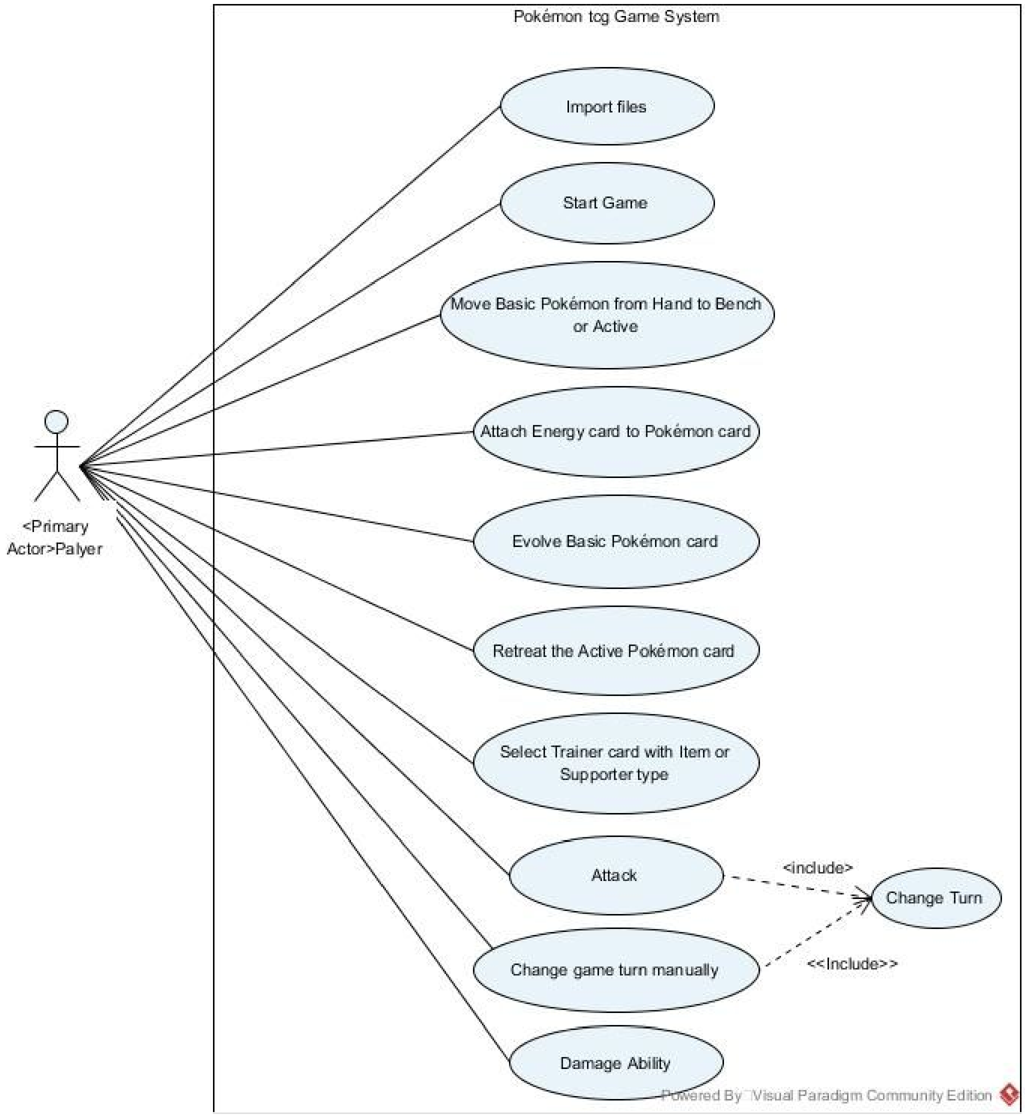

Pokémon tcg Game System
Import files
Start Game
Move Basic Pokémon from Hand to Bench or Active
Attach Energy card to Pokémon card
Evolve Basic Pokémon card
Retreat the Active Pokémon card
Select Trainer card with Item or Supporter type
Attack
<include>
Change Turn
Change game turn manually
<<Include>>
Damage Ability
<Primary Actor>Palyer
Powered By Visual Paradigm Community Edition

Figure SEQ Figure * ARABIC 1. Main use case of PokémonGoBack System

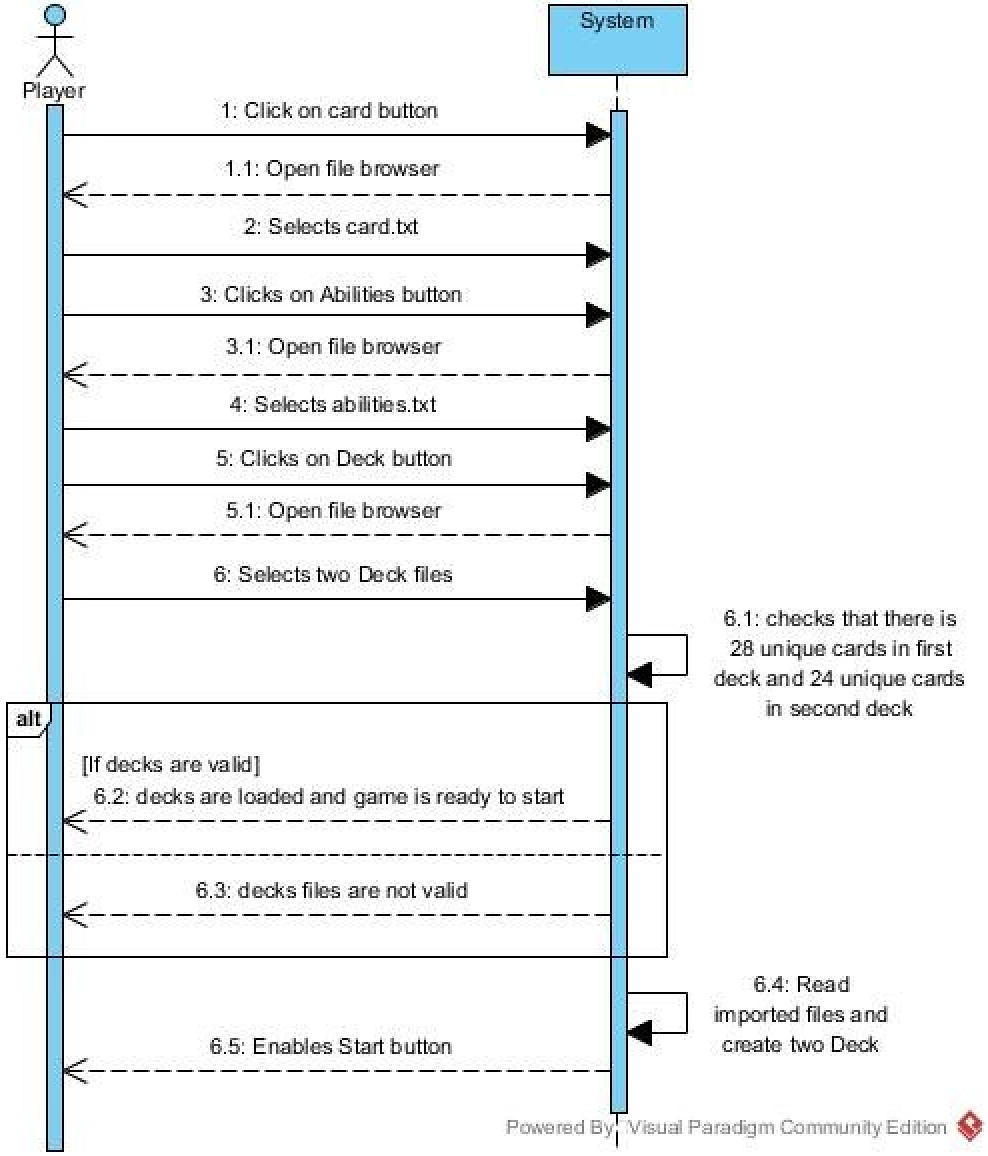
2.1 Use Case Description-UC01- Import Deck, Abilities, and Card files

Table 1UC01-import Files

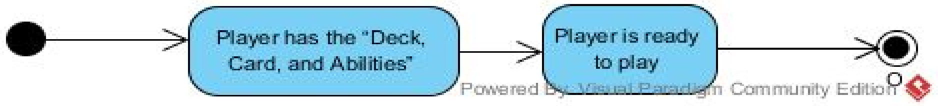
Use-Case ID:	UC01		
Use-Case Name:	Import Deck, Abilities, and Card files.		
Date Created:	05/07/2017	**Date Last Updated:**	05/16/2017

Description:	This use case allows Player to import Deck, Abilities, and Card files which is generated from Pokémon tcg online.
Priority:	High
Primary Actor:	Player
Secondary Actor:	None.
Trigger:	Player clicks on the “Deck, Card, or Abilities” button before starting the game.
Preconditions:	The “Deck, Card, and Abilities” buttons are active before starting the game.
Post conditions:	Files were loaded, and game is ready to start.
Normal Flow:	1. Player clicks on the “Card” button. 2. System opens the file browser to select the card.text file. 3. Player selects card file. 4. Player clicks on the “Abilities” button. 5. System opens the file browser to select the abilities.text file. 6. Player selects Abilities file. 7. Player clicks on “Deck” button. 8. System opens the file browser to select the deck.text files. 9. Player selects two Deck files. 10. System checks that there is no 28 unique cards in first Deck and 24 unique cards in second Deck. 11. System read imported files and create two Decks. 12. System enables the “Start” button.
Exceptions Flow:	10.a There is no 28 unique cards in first Deck or 24 unique cards in second Deck. System announces that imported Deck files are not valid. System stops the importing process.
Includes:	None.
Notes and Issues:	None.

2.1.1 SSD-UC01-Import Deck, Abilities, and Card files


Player
System
1: Click on card button
1.1: Open file browser
2: Selects card.txt
3: Clicks on Abilities button
3.1: Open file browser
4: Selects abilities.txt
5: Clicks on Deck button
5.1: Open file browser
6: Selects two Deck files
6.1: checks that there is 28 unique cards in first deck and 24 unique cards in second deck
alt
[If decks are valid]
6.2: decks are loaded and game is ready to start
6.3: decks files are not valid
6.4: Read imported files and create two Deck
6.5: Enables Start button
Powered By: Visual Paradigm Community Edition

2.1.2 Activity Diagram-UC01-Import files


Player has the "Deck, Card, and Abilities"
Player is ready to play
Powered By: Visual Paradigm Community Edition

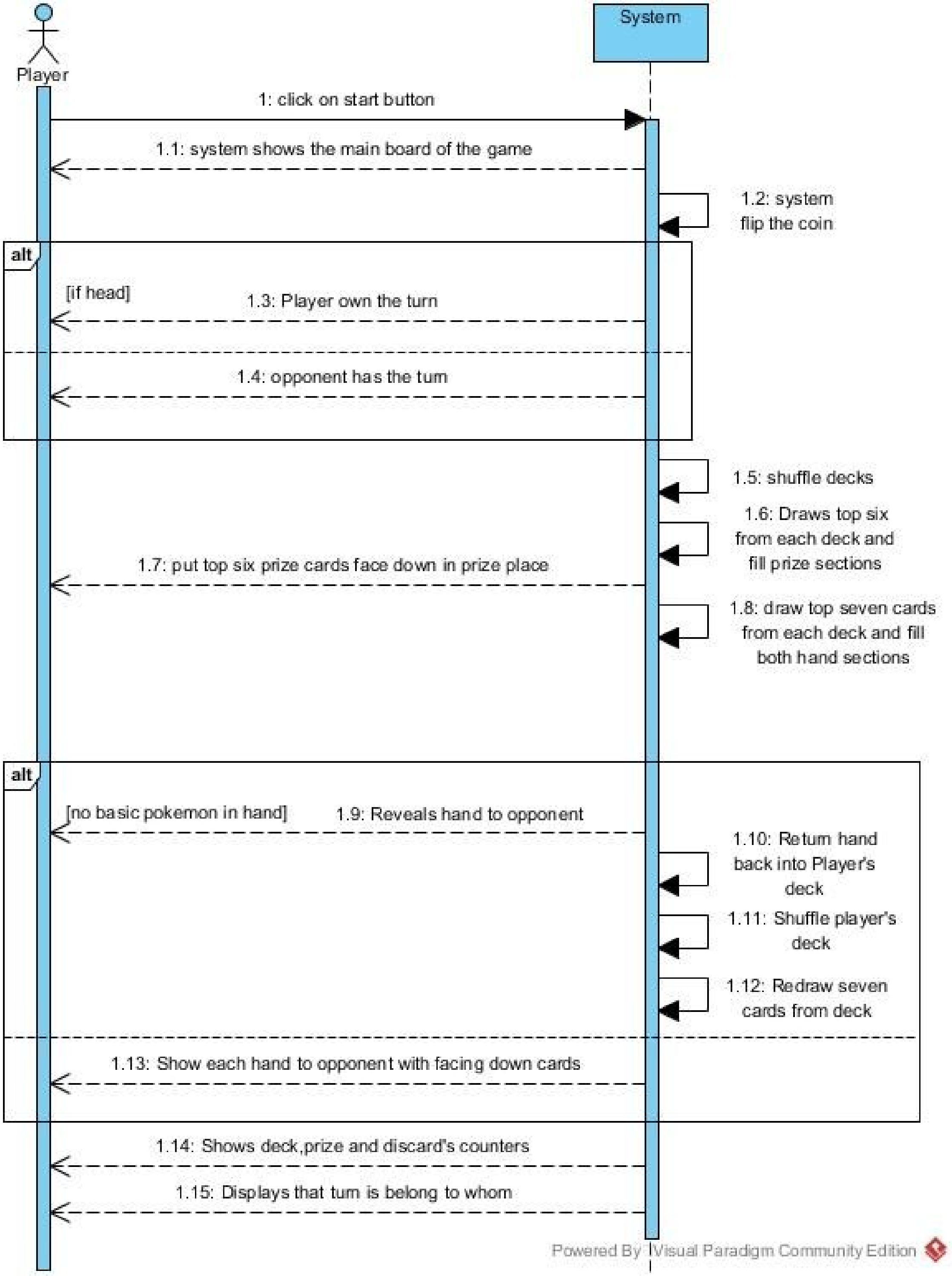
2.2 Use Case Description-UC02-Start Game

Table 2UC02-Start Game

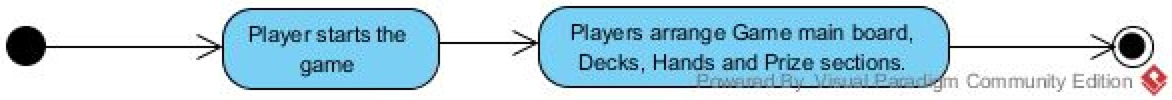
Use-Case ID:	UC02		
Use-Case Name:	Start Game.		
Date Created:	05/07/2017	**Date Last Updated:**	05/18/2017

Description:	This use case allows Player to start the game.
Priority:	High
Primary Actor:	Player
Secondary Actor:	None
Trigger:	Player clicks on the "Start" button for starting the game.
Preconditions:	● Card, Abilities and two Deck files were imported by user.
Post conditions:	● Game main board, Decks, Hands and Prize sections were loaded. ● The game is ready to play.
Normal Flow:	1. Player clicks on the "Start" button. 2. System shows the main board of the game. 3. System flips a coin and decides who goes first. 4. System shuffles both Decks which are filled with decks information. 5. System draws top six cards from each Deck and fills both Prize sections. 6. System puts the top six Prize cards face down in the Prize place. 7. System draws top seven cards from each Deck and fill both Hand sections. 8. System shows each Hand to the Opponent with facing down cards. 9. System shows the Deck, Prize and Discards' counters. 10. System shows that turn is belong to whom.
Exceptions Flow:	7.a There is no Basic Pokémon in Hand: System reveals Hand to Opponent. System returns Hand back into Player's Deck. System shuffles Player's Deck. System redraws seven cards from the Deck. If Player still doesn't have any Basic Pokémon, system repeats it.
Includes:	None.
Notes and Issues:	None.

2.2.1 SSD-UC02-Start Game


Player
System
1: click on start button
1.1: system shows the main board of the game
1.2: system flip the coin
alt
[if head]
1.3: Player own the turn
1.4: opponent has the turn
1.5: shuffle decks
1.6: Draws top six from each deck and fill prize sections
1.7: put top six prize cards face down in prize place
1.8: draw top seven cards from each deck and fill both hand sections
alt
[no basic pokemon in hand]
1.9: Reveals hand to opponent
1.10: Return hand back into Player's deck
1.11: Shuffle player's deck
1.12: Redraw seven cards from deck
1.13: Show each hand to opponent with facing down cards
1.14: Shows deck,prize and discard's counters
1.15: Displays that turn is belong to whom
Powered By Visual Paradigm Community Edition

2.2.2 Activity Diagram-UC02-Start Game


Player starts the game
Players arrange Game main board, Decks, Hands and Prize sections.
Powered By Visual Paradigm Community Edition

2.3 Use Case Description-UC03-Move Basic Pokémon from Hand to Bench or Active

Table 3UC03- Move Basic Pokémon from Hand to Bench or Active

Use-Case ID:	UC03		
Use-Case Name:	Move Basic Pokémon from Hand to Bench or Active.		
Date Created:	05/07/2017	**Date Last Updated:**	05/16/2017

Description:	This use case allows the Player to move Basic Pokémon from Hand to Active or Bench.
Priority:	High
Primary Actor:	Player
Secondary Actor:	None
Trigger:	Player clicks on one Basic Pokémon in Hand.
Preconditions:	● There is at least one Basic Pokémon in Hand. ● There is less than 5 Pokémon cards in Bench. Or Active place is free.
Post conditions:	The selected Basic Pokémon moved to Active or Bench.
Normal Flow:	1. Player clicks on one Basic Pokémon on Hand. 2. System shows the selected card information in illustrator. 3. Player clicks on a valid location (Bench or Active) to move Pokémon.
Exceptions Flow:	1.a Player selects Trainer or Energy card and moves it to Active or Bench: System shows a message: “Selected card is not valid to move to Active or Bench!” 3.a Player clicks on an invalid location to move: System shows a message: “Selected destination is not a valid location!”
Includes:	None.
Notes and Issues:	None.

2.3.1 SSD-UC03- Move Basic Pokémon from Hand to Bench or Active

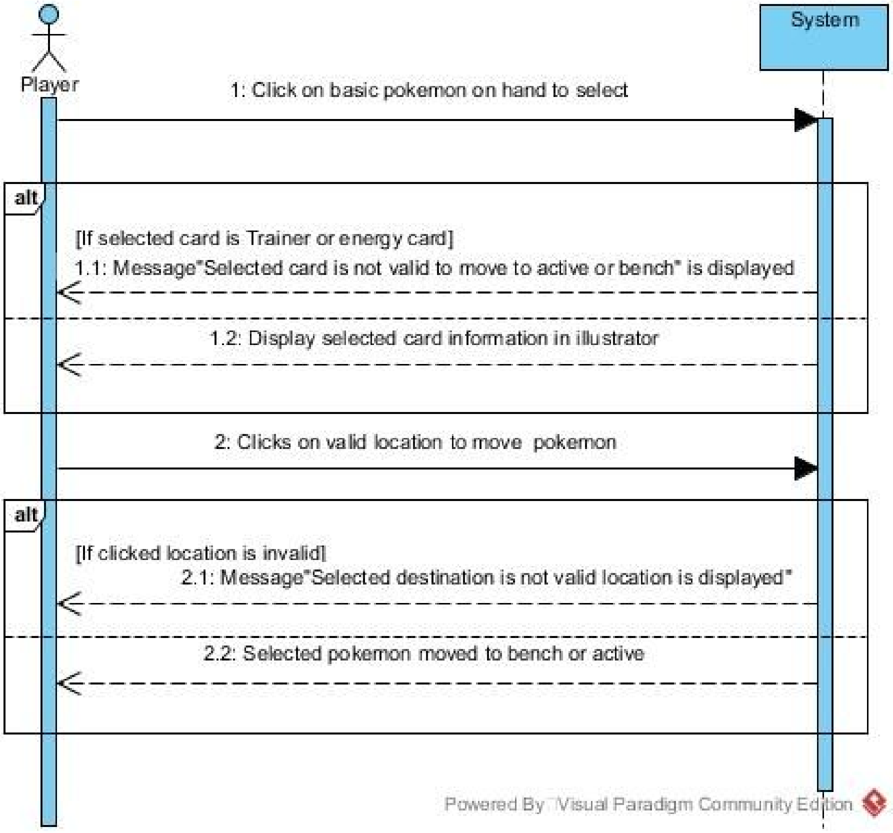
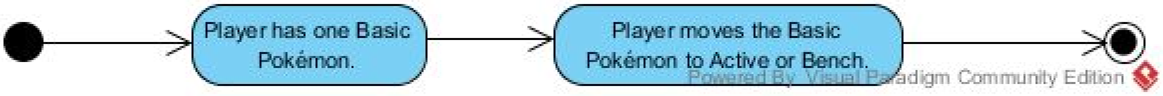

Player
System
1: Click on basic pokemon on hand to select
alt
[If selected card is Trainer or energy card]
1.1: Message"Selected card is not valid to move to active or bench" is displayed
1.2: Display selected card information in illustrator
2: Clicks on valid location to move pokemon
alt
[If clicked location is invalid]
2.1: Message"Selected destination is not valid location is displayed"
2.2: Selected pokemon moved to bench or active
Powered By Visual Paradigm Community Edition

Figure SEQ Figure * ARABIC 6. SSD-UC03- Move Basic Pokémon from Hand to Bench or Active

2.3.2 Activity Diagram-UC03- Move Basic Pokémon from Hand to Bench or Active


Player has one Basic Pokémon.
Player moves the Basic Pokémon to Active or Bench.
Powered By Visual Paradigm Community Edition

2.4 Use Case Description-UC04- Attach an Energy card to one Pokémon card

Table 4UC04 Attach an Energy card to one Pokémon card

Use-Case ID:	UC04		
Use-Case Name:	Attach an Energy card to one Pokémon card.		
Date Created:	05/07/2017	**Date Last Updated:**	05/18/2017

Description:	This use case allows the Player to attach Energy to Active Pokémon or one of his Benched Basic or Evolved Pokémon just once each turn.

Priority:	High
Primary Actor:	Player
Secondary Actor:	None
Trigger:	Player takes one type of Energy card from his Hand.
Preconditions:	● Player hasn't attach Energy to Pokémon in this turn. ● Player has one or more Energy cards in his Hand. ● Active Pokémon is available in the players' Active place.
Post conditions:	● The Pokémon got Energy. ● The attached Energy showed on bottom of the Pokémon card.
Normal Flow:	1. Player selects Energy card from his Hand. 2. System shows selected card in illustrator. 3. Player clicks on a valid location to attach Energy to Pokémon card. 4. System shows Attached Energy on bottom of the Pokémon card.
Exceptions Flow:	3.a Player clicks on an invalid location to attach Energy: System shows a message: "Selected destination is not a valid location!"
Includes:	None.
Notes and Issues:	None.

2.4.1 SSD-UC04-Attach an Energy card to one Pokémon card

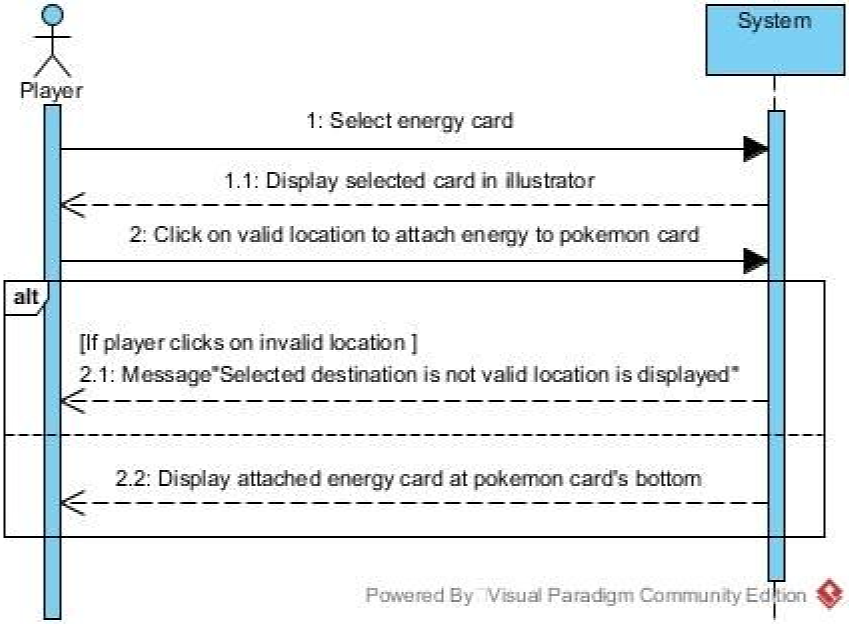
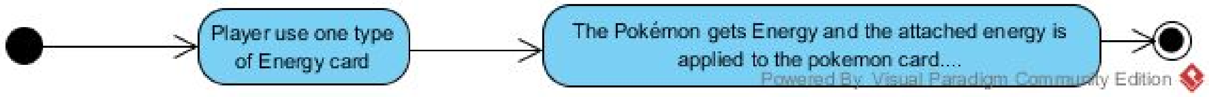

Player
System
1: Select energy card
1.1: Display selected card in illustrator
2: Click on valid location to attach energy to pokemon card
alt
[If player clicks on invalid location]
2.1: Message"Selected destination is not valid location is displayed"
2.2: Display attached energy card at pokemon card's bottom
Powered By Visual Paradigm Community Edition

Figure SEQ Figure * ARABIC 8. SSD-UC04-Attach an Energy card to one Pokémon card

2.4.2 Activity Diagram-UC04-Attach an Energy card to one Pokémon card


Player use one type of Energy card
The Pokémon gets Energy and the attached energy is applied to the pokemon card....
Powered By Visual Paradigm Community Edition

2.5 Use Case Description-UC05-Evolve Basic Pokémon card

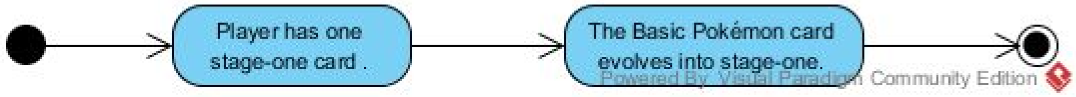
Table 5UC05- Evolve Basic Pokémon card

Use-Case ID:	UC05		
Use-Case Name:	Evolve Basic Pokémon card.		
Date Created:	05/07/2017	**Date Last Updated:**	05/18/2017

Description:	This use case allows the Player to Evolve Basic Pokémon card which is at Active or Bench place into stage 1.
Priority:	High
Primary Actor:	Player
Secondary Actor:	None
Trigger:	Player selects one stage-one card from his Hand.
Preconditions:	● There is at least one stage-one card in Hand. ● Active Pokémon is available in the players' Active place. ● It's the first time in this turn that Player evolves the Basic Pokémon. So, Player can't evolve it a second time the same turn.
Post conditions:	● The Basic Pokémon card evolved into stage-one. It kept all cards attached to it (Energy, etc.) and any damage counters on it. Any effects of attacks affecting the Pokémon—such as Asleep, Confused, or Poisoned—end when it evolved.
Normal Flow:	1. Player selects one stage-one card from his Hand. 2. System shows card information in illustrator. 3. Player clicks on a valid location (Basic Active or Benched Pokémon) to Evolve. 4. System keeps all attached card (Energy card) and damage counters on it. 5. System removes any effects of attack. 6. System shows "Stage-one" on top of the card.
Exceptions Flow:	3.a If Player wants to move invalid card (for instance, selected card is a Trainer card) to Basic Active or Benched Pokémon: System shows a message: "Selected card is invalid to Evolve!" 3.b If Player clicks on an invalid location: System shows a message: "Selected destination is not a valid location!"
Includes:	None.
Notes and Issues:	None.

2.5.1 SSD-UC05- Evolve Basic Pokémon card

2.5.2 Activity Diagram-UC05- Evolve Basic Pokémon card


Player has one stage-one card .
The Basic Pokémon card evolves into stage-one.
Powered By Visual Paradigm Community Edition

2.6 Use Case Description-UC06-Use Ability

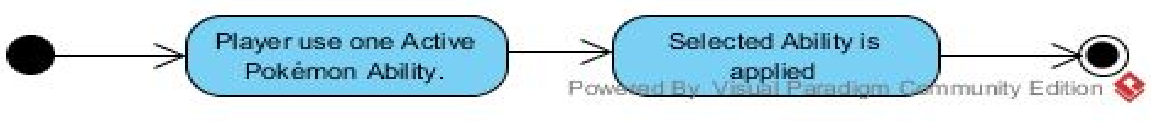
Table 6UC06 Use Ability

Use-Case ID:	UC06		
Use-Case Name:	Use Ability		
Date Created:	05/07/2017	**Date Last Updated:**	05/18/2017

Description:	This use case allows user (Player or Opponent) to use his Active Pokémon Ability.
Priority:	High
Primary Actor:	Player
Secondary Actor:	None.
Trigger:	Player selects one Active Pokémon Ability.
Preconditions:	● Active Pokémon is available in the players' Active place. ● The Active Pokémon has special Ability. ● Some Abilities work only if a condition is met, while others work all the time even without player using them. ● Player in this turn hasn't Attack yet.
Post conditions:	Selected Ability was applied.
Normal Flow:	1. Player clicks on the Active Pokémon. 2. System shows card information in Illustrator. 3. Player clicks on "Apply Ability" button. 4. System checks conditions is met before using Ability. 5. System announces which Ability Player is using. Figure SEQ Figure * ARABIC 111. Activity Diagram-UC05- Evolve Basic Pokémon card 6. System applies the Ability scenario on the target. 7. System shows the effect of the Ability.
Exceptions Flow:	None.
Includes:	None.
Notes and Issues:	None.

2.6.1 SSD-UC06- Use Ability

2.6.2 Activity Diagram-UC06-Use Ability


Player use one Active Pokémon Ability.
Selected Ability is applied
Powered By Visual Paradigm Community Edition

2.7 Use Case Description-UC07- Retreat the Active Pokémon card

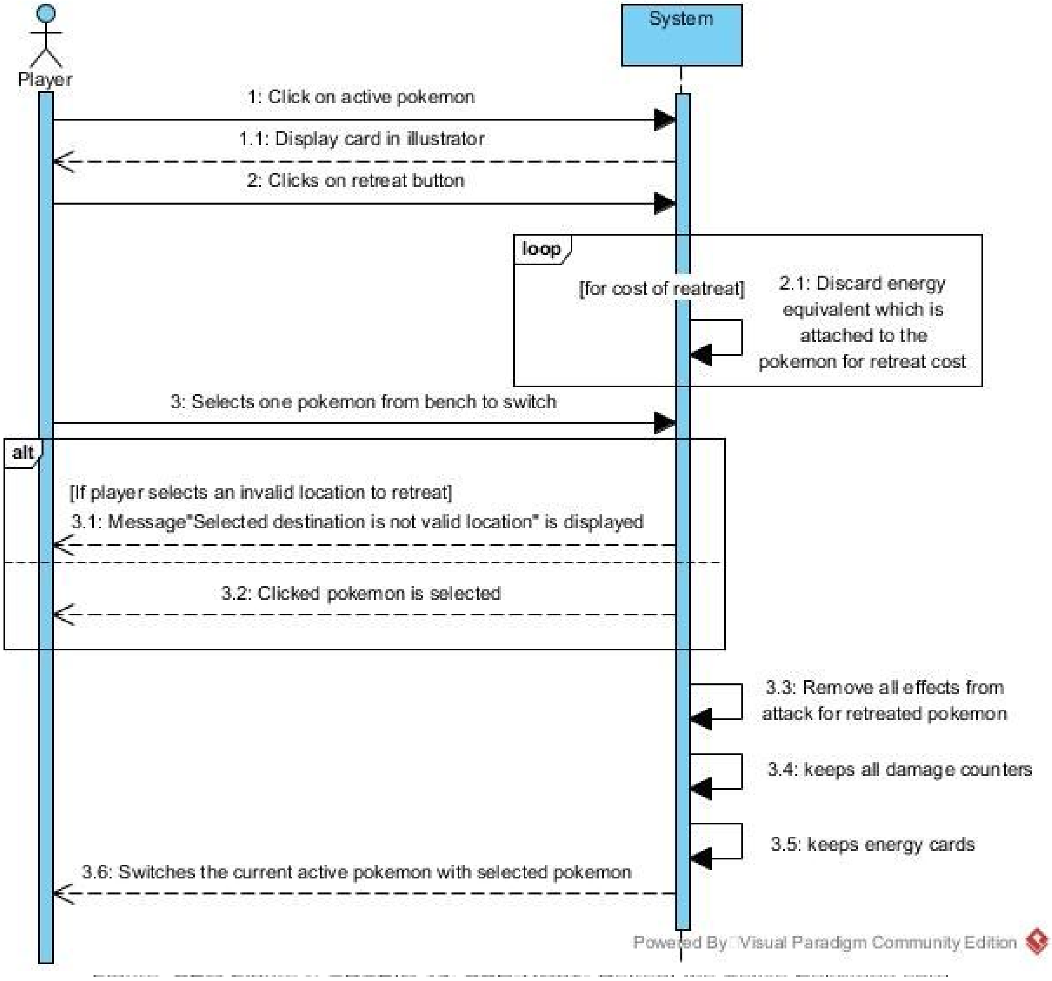
Table 7UC07- Retreat the Active Pokémon card

Use-Case ID:	UC07		
Use-Case Name:	Retreat the Active Pokémon card.		
Date Created:	05/07/2017	**Date Last Updated:**	05/18/2017

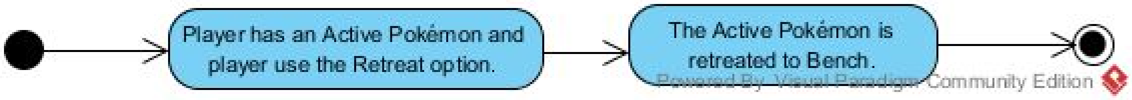
Description:	This use case allows the Player to Retreat Active Pokémon once per turn when there are lots of damage counters on it or there is a strong Pokémon on the Bench that readies to battle.
Priority:	High
Primary Actor:	Player
Secondary Actor:	None
Trigger:	Player clicks on Active Pokémon and selects the Retreat option.
Preconditions:	● Player hasn't retreat in this turn (it is first time in this turn). ● Active Pokémon that are Asleep or Paralyzed cannot retreat. ● Active Pokémon is available in the players' Active place.
Post conditions:	● The Active Pokémon retreated to Bench. ● Any effects from attacks goes away. ● Player must discard 1 Energy from Active Pokémon for each ★ listed in its Retreat Cost. If no ★ are listed, it retreats for free. ● All damage counters and attached cards kept. ● Player can still attack with new Active Pokémon.
Normal Flow:	1. Player clicks on Active Pokémon. 2. System shows card in illustrator. 3. Player clicks on "Retreat" button. 4. System discards 1 Energy from Active Pokémon for each ★ listed in its Retreat Cost. 5. Player selects one Pokémon from Bench to switch. 6. System removes all effects from Attacks for the current Pokémon. 7. System switches the current Active Pokémon with the selected Pokémon from Player's Bench. 8. System keeps all damage counters and attached cards for retreated Pokémon.
Exceptions Flow:	4. a If no ★ are listed: It retreats for free.

	5.a Player selects an invalid location to Retreat: System shows a message: “Selected destination is not a valid location!”
Includes:	None.
Notes and Issues:	None.

2.7.1 SSD-UC07- Retreat the Active Pokémon card


Player
System
1: Click on active pokemon
1.1: Display card in illustrator
2: Clicks on retreat button
loop
[for cost of reatreat]
2.1: Discard energy equivalent which is attached to the pokemon for retreat cost
3: Selects one pokemon from bench to switch
alt
[If player selects an invalid location to retreat]
3.1: Message"Selected destination is not valid location" is displayed
3.2: Clicked pokemon is selected
3.3: Remove all effects from attack for retreated pokemon
3.4: keeps all damage counters
3.5: keeps energy cards
3.6: Switches the current active pokemon with selected pokemon
Powered By Visual Paradigm Community Edition

2.7.2 Activity Diagram-UC07- Retreat the Active Pokémon card


Player has an Active Pokémon and player use the Retreat option.
The Active Pokémon is retreated to Bench.
Powered By Visual Paradigm Community Edition

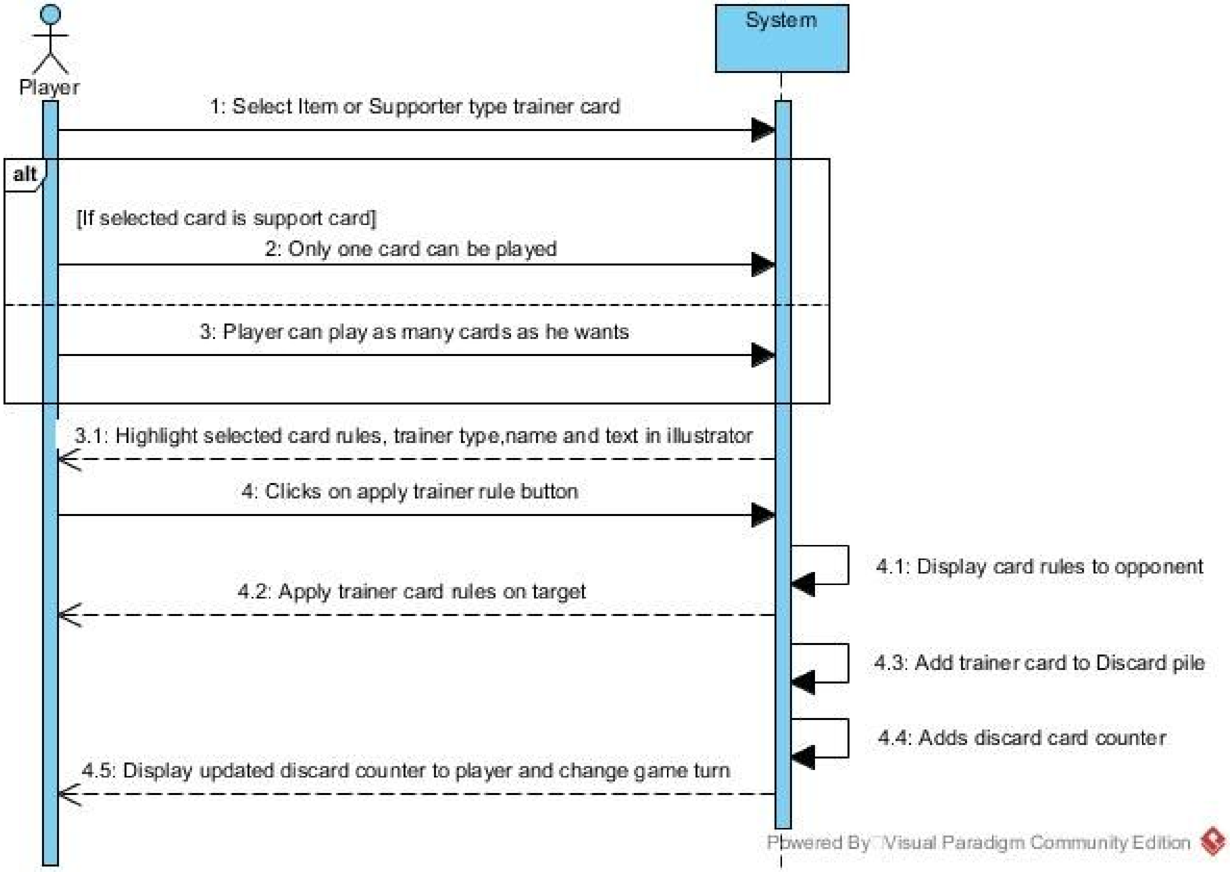
2.8 Use Case Description-UC08- Select Item or Supporter Trainer card

Table 8UC08- Select Item or Supporter Trainer card

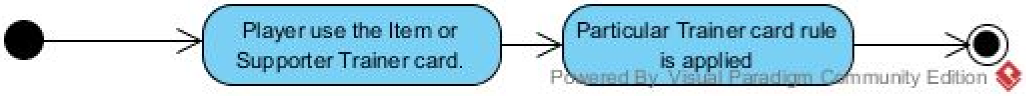
Use-Case ID:	UC08		
Use-Case Name:	Select Item or Supporter Trainer card.		
Date Created:	05/07/2017	**Date Last Updated:**	05/18/2017

Description:	This use case allows user (Player or Opponent) to select a Trainer card with Item or Supporter type and obeys their rules. Only one Supporter card can use per turn. Item card can play as many as he like.
Priority:	High
Primary Actor:	Player
Secondary Actor:	None.
Trigger:	Player clicks on Item or Supporter Trainer card.
Preconditions:	● There is at least one Trainer card with Item or Supporter type in Hand. ● The game turn has to belong to the player. ● The Supporter card hasn't use yet in this turn. ● Active Pokémon is available in the players' Active place.
Post conditions:	● Selected Trainer card rule was applied. ● Selected Trainer card put in the Discard pile.
Normal Flow:	1. Player selects a Trainer card with Item or Supporter type. 2. System shows the Item card rules at the bottom of the card, Trainer type, Trainer name, and Trainer text to the Player in illustrator. 3. Player clicks on "Apply Trainer Rule" button. 4. System shows the Trainer card rules to the Opponent. 5. System applies the rules of the Trainer card on its target which is defined on the rules. 6. System adds the Trainer card to the Discard pile. 7. System adds the Discards card counter. 8. System changes the game turn.
Exceptions Flow:	1.a If selected card is a Supporter card: Player can play only one Supporter card each turn. 1.b If selected card is an Item card: Player can play as many Item cards as he like.
Includes:	None.
Notes and Issues:	None.

2.8.1 SSD-UC08- Select Item or Supporter Trainer card


Player
System
1: Select Item or Supporter type trainer card
alt
[If selected card is support card]
2: Only one card can be played
3: Player can play as many cards as he wants
3.1: Highlight selected card rules, trainer type,name and text in illustrator
4: Clicks on apply trainer rule button
4.1: Display card rules to opponent
4.2: Apply trainer card rules on target
4.3: Add trainer card to Discard pile
4.4: Adds discard card counter
4.5: Display updated discard counter to player and change game turn
Powered By Visual Paradigm Community Edition

2.8.2 Activity Diagram-UC08- Select Item or Supporter Trainer card


Player use the Item or Supporter Trainer card.
Particular Trainer card rule is applied
Powered By Visual Paradigm Community Edition

2.9 Use Case Description-UC09-Attack

Table 9UC09-Attack

Use-Case ID:	UC09		
Use-Case Name:	Attack		
Date Created:	05/07/2017	**Date Last Updated:**	5/18/2017

Description:	This use case allows Player to attack to the Opponent Active Pokémon.
Priority:	High
Primary Actor:	Player
Secondary Actor:	None.
Trigger:	Player selects one Active Pokémon.
Preconditions:	● Active Pokémon is available in the players' Active place.

	● The Active Pokémon has to have right amount Energy to Attack. ● The Active Pokémon has not to be Asleep or Paralyzed. ● The Active Pokémon has not attack in this turn. ● if player wants to move Pokémon from hand to Bench, attach Energies, evolve Pokémon, Retreat Active Pokémon, use Ability, and use Trainer card, they has to finish them before attacking.
Post conditions:	● Once Player attacked, and Player turn was over. ● The effect of Attack showed on Opponent Active Pokémon. ● If Player knocked out the Opponent, the Prize list showed to Player.
Normal Flow:	1. Player clicks on the Active Pokémon. 2. System highlights the Attack options which have right amount of Energy for Attack. 3. Player clicks on “Attack1 or Attack2” button. 4. System reads Weakness and Resistance of the target Pokémon. 5. System reads the amount damage of Attack. 6. If the energy type of selected action is the same as weakness energy type, system multiply the action damage with weakness factor which is defined in opponent active Pokémon weakness. System consider this number as damage count. 7. System checks to see if any Pokémon were Knocked Out by the attack. 8. System checks if total damage of targeting Pokémon is at least equal to its HP, it is Knocked Out. 9. If Player knocked out the Opponent, his Pokémon and all cards attached to it goes to the Discard pile 10. If Player knocked out the Opponent and there is more than one card in Prize list: System shows the prize list (face down) to Player. 11. If there is only one prize card: System announces the Player wins. 12. If Player doesn’t knock out the Opponent, Player turn is over. 13. System gives the turn to the Opponent.
Exceptions Flow:	1. a There is no Active Pokémon on the players’ Active place. System shows a message: “Move Pokémon to Active section!” 3. a The Active Pokémon has not enough Energy for none of its Attack options. If Player clicks on “Attack1 or Attack2”button: System shows a message: “Pokémon doesn’t have enough energy to Attack!” 2.b If Active Pokémon has this cost symbol ✪. That means the attack has a cost of 0, Player can use it without any Energy attached to the Pokémon! 8.a System checks to make sure every Pokémon that was affected by the attack. Some attacks can damage more than one Pokémon, and sometimes they can even damage the Attacking Pokémon.
Includes:	Use-case Change Turn scenario.
Notes and Issues:	None.

2.9.1 SSD-UC09- Attack

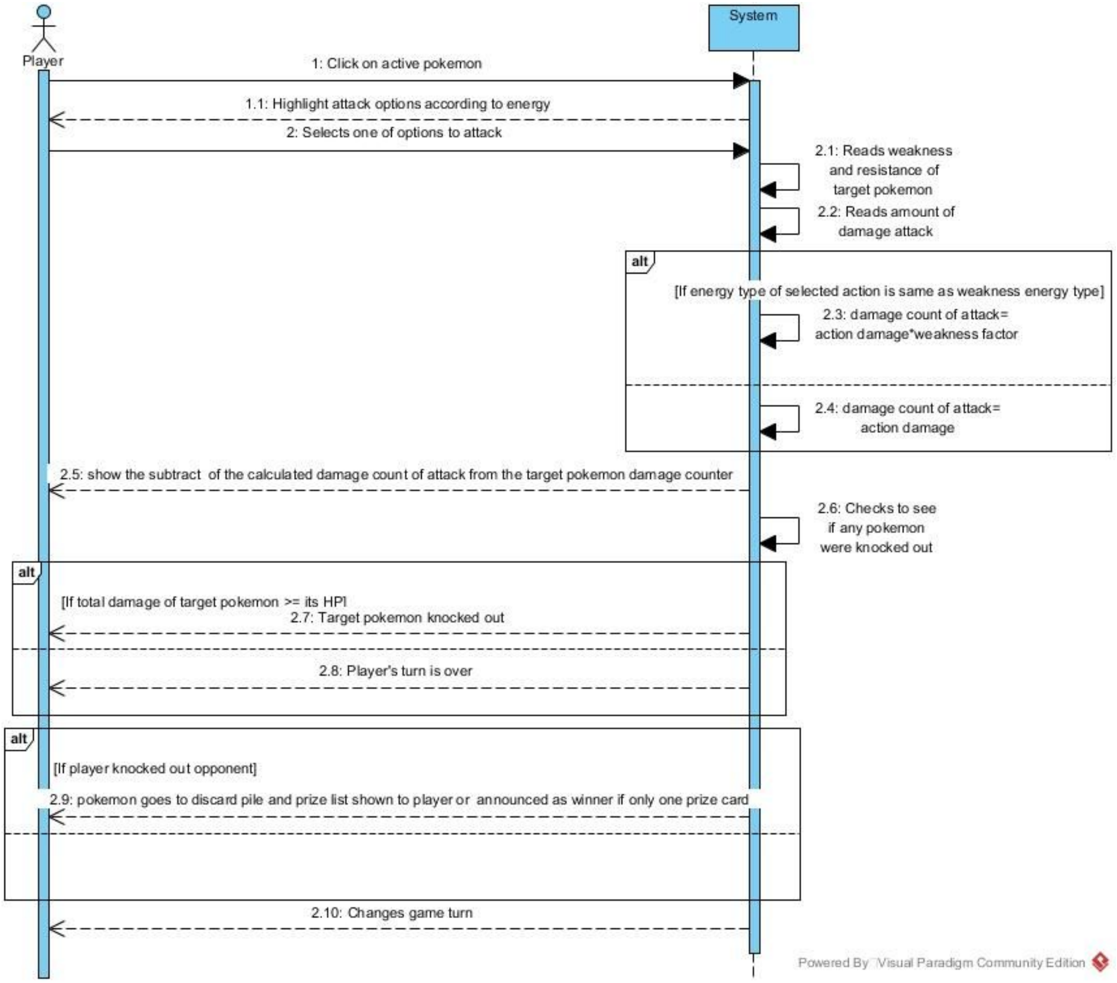
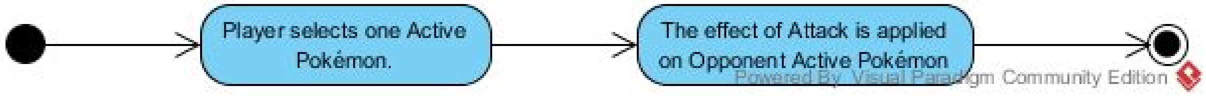

Player
System
1: Click on active pokemon
1.1: Highlight attack options according to energy
2: Selects one of options to attack
2.1: Reads weakness and resistance of target pokemon
2.2: Reads amount of damage attack
alt
[If energy type of selected action is same as weakness energy type]
2.3: damage count of attack= action damage*weakness factor
2.4: damage count of attack= action damage
2.5: show the subtract of the calculated damage count of attack from the target pokemon damage counter
2.6: Checks to see if any pokemon were knocked out
alt
[If total damage of target pokemon >= its HP]
2.7: Target pokemon knocked out
2.8: Player's turn is over
alt
[If player knocked out opponent]
2.9: pokemon goes to discard pile and prize list shown to player or announced as winner if only one prize card
2.10: Changes game turn
Powered By Visual Paradigm Community Edition

Figure SEQ Figure * ARABIC 18. SSD-UC09-Attack

2.9.2 Activity Diagram-UC09-Attack


Player selects one Active Pokémon.
The effect of Attack is applied on Opponent Active Pokémon
Powered By Visual Paradigm Community Edition

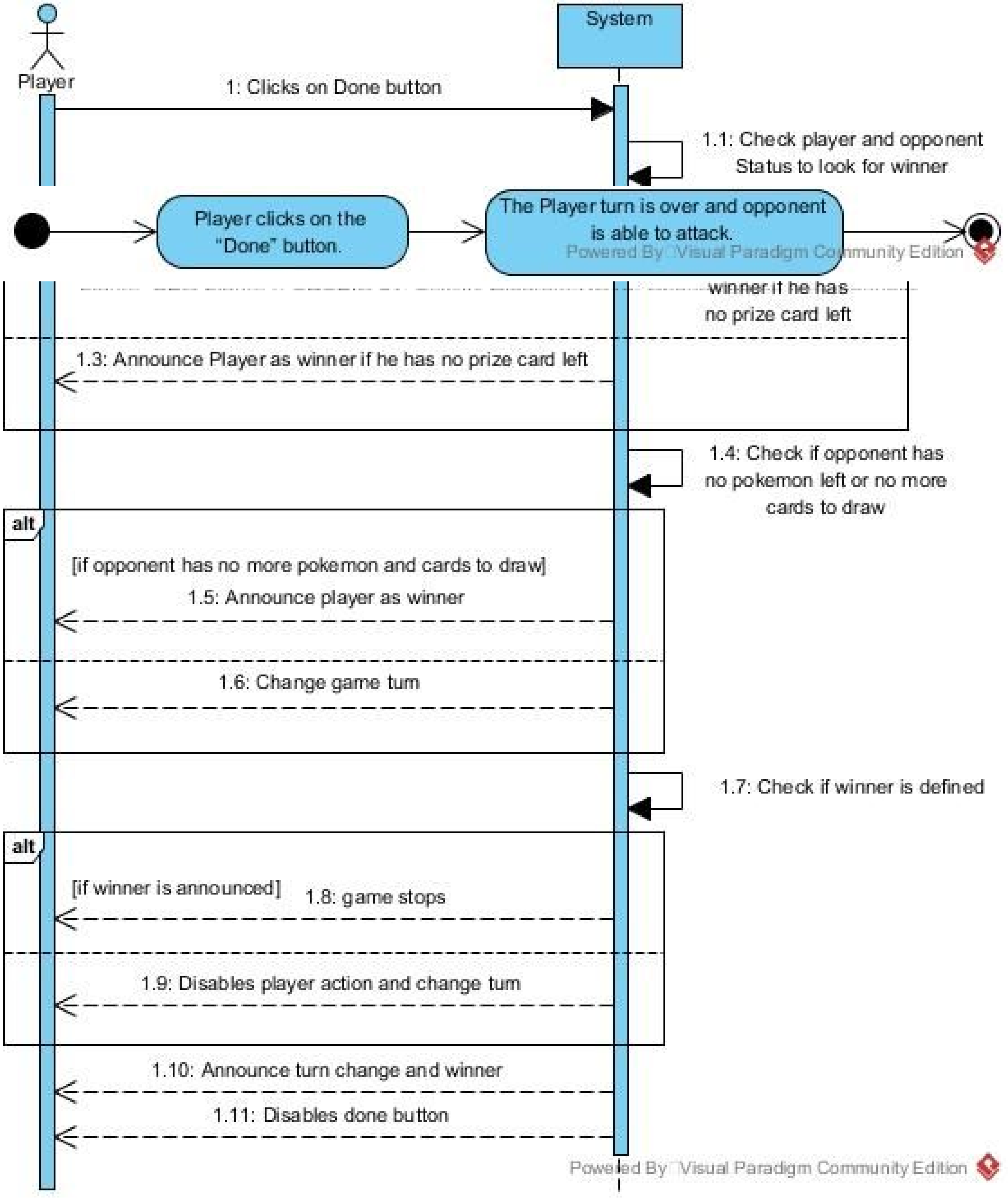
2.10 Use Case Description-UC10-Change game turn manually

Table 10UC10- Change game turn manually

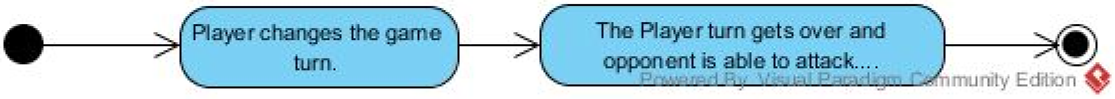
Use-Case ID:	UC10		
Use-Case Name:	Change game turn manually.		
Date Created:	05/07/2017	**Date Last Updated:**	05/18/2017

Description:	This use case allows Player to change the game turn to the Opponent manually whenever he wants.
Priority:	High
Primary Actor:	Player
Secondary Actor:	None.
Trigger:	Player clicks on the “Done” button.
Preconditions:	Done Button is activated during the player turn.
Post conditions:	● The Player turn was over. ● Opponent be able to act. ● The Player “Done” button is disabled.
Normal Flow:	1. Player clicks on the “Done” button. 2. System checks the Player and Opponent status in order to understand is there any winner at the changing turn action 3. System checks the Opponents’ Prize cards. If it is zero, system announces that the Opponent is winner. 4. System checks the Players’ Prize cards. If it is zero, system announces that the Player is winner. 5. System checks whether the Opponent has no Pokémon left in play, or no cards left to draw at the beginning of the changing the turn. If so, system announces the Player as a winner. 6. System checks whether the winner is defined. If winner is announced the game stops. 7. If there is no winner, at this stage, system disables all player action then active Opponent action. 8. System announces that turn is changed and defined who the owner of the turn is. 9. System disables the “Done” button.
Exceptions Flow:	None.
Includes:	None.
Notes and Issues:	None.

2.10.1 SSD-UC10- Change game turn manually


Player
System
1: Clicks on Done button
1.1: Check player and opponent Status to look for winner
Player clicks on the "Done" button.
The Player turn is over and opponent is able to attack.
Powered By Visual Paradigm Community Edition
winner if he has no prize card left
1.3: Announce Player as winner if he has no prize card left
1.4: Check if opponent has no pokemon left or no more cards to draw
alt
[if opponent has no more pokemon and cards to draw]
1.5: Announce player as winner
1.6: Change game turn
1.7: Check if winner is defined
alt
[if winner is announced]
1.8: game stops
1.9: Disables player action and change turn
1.10: Announce turn change and winner
1.11: Disables done button
Powered By Visual Paradigm Community Edition

2.10.2 Activity Diagram-UC10- Change game turn manually


Player changes the game turn.
The Player turn gets over and opponent is able to attack....
Powered By Visual Paradigm Community Edition

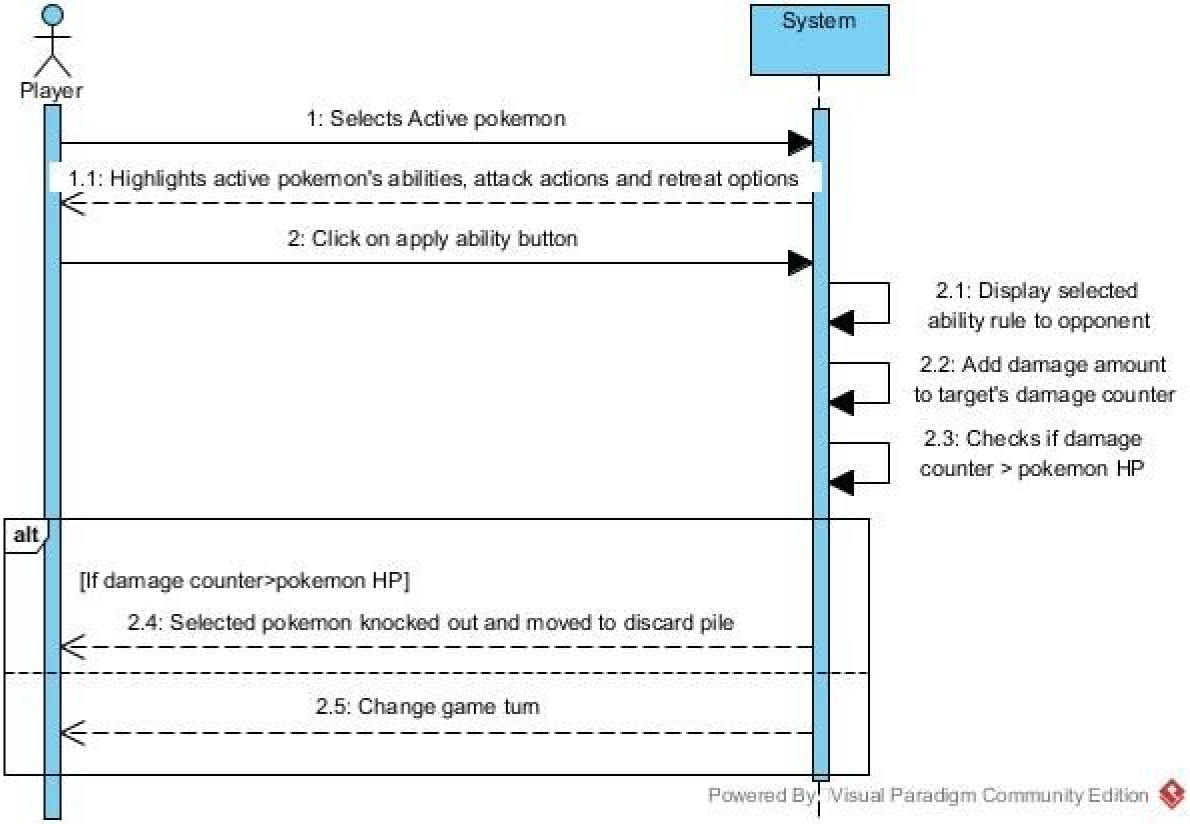
2.11 Use Case Description-UC11-Damage Ability

Table 11 UC11-Damage Ability

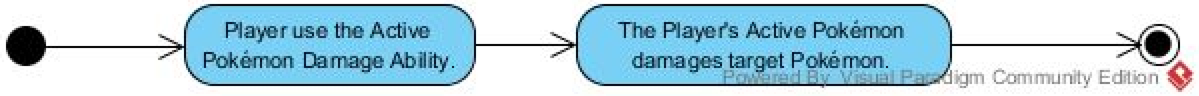
Use-Case ID:	UC11		
Use-Case Name:	Damage Ability		
Date Created:	05/07/2017	**Date Last Updated:**	05/18/2017

Description:	This use case allows the Player to use Damage Ability in order to damage the target Pokémon.
Priority:	High
Primary Actor:	Player
Secondary Actor:	None.
Trigger:	Player clicks on the Active Pokémon Damage Ability.
Preconditions:	● Active Pokémon is available in the players' Active place. ● The Active Pokémon has Damage Ability. ● Player in this turn hasn't Attack yet. ● Opponent has an Active Pokémon.
Post conditions:	● The Player Active Pokémon damaged target Pokémon.
Normal Flow:	1. Player selects Active Pokémon. 2. System highlights Active Pokémon Abilities, Attack actions, and Retreat option. 3. Player clicks on "Apply Ability" button. 4. System shows selected abilities rule to the Opponent. 5. System adds the damage amount to the target Pokémon damage counter. 6. System checks if damage counter is more than its HP, system announces the selected Pokémon is knocked out. 7. System moves knocked out Pokémon and all cards attached to it to the Discard pile.
Exceptions Flow:	5.a if target is a Pokémon on the Bench: Player selects one Pokémon on the target Bench.
Includes:	None.
Notes and Issues:	None.

2.11.1 SSD-UC11-Damage Ability


Player
System
1: Selects Active pokemon
1.1: Highlights active pokemon's abilities, attack actions and retreat options
2: Click on apply ability button
2.1: Display selected ability rule to opponent
2.2: Add damage amount to target's damage counter
2.3: Checks if damage counter > pokemon HP
alt
[If damage counter>pokemon HP]
2.4: Selected pokemon knocked out and moved to discard pile
2.5: Change game turn
Powered By Visual Paradigm Community Edition

2.11.2 Activity Diagram-UC11-Damage Ability


Player use the Active Pokémon Damage Ability.
The Player's Active Pokémon damages target Pokémon.
Powered By Visual Paradigm Community Edition

3

GUI Decision

3.1 Import Files

User Clicks on these three files, figure 24, then system opens file Brower, figure 25. User select files. After importing "Start Game" button is enabled figure26.

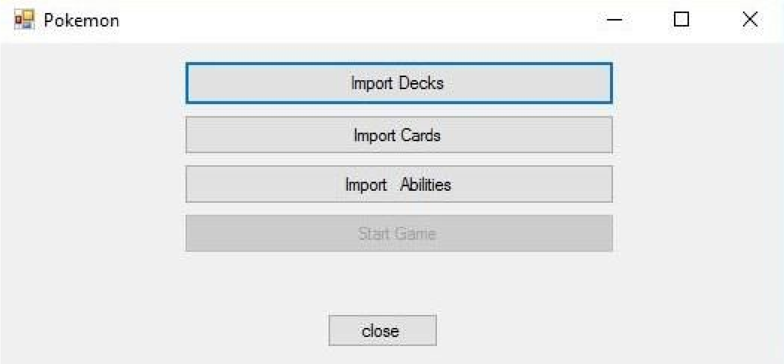
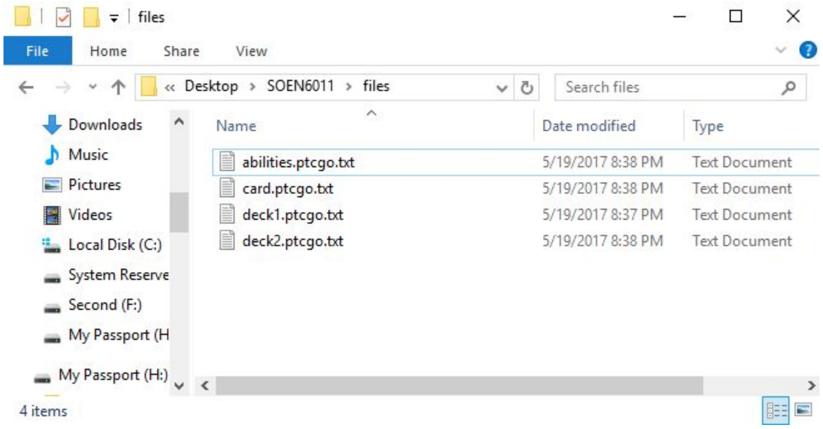


Figure SEQ Figure * ARABIC 26.First Page-Import Files



Name	Date modified	Type
abilities.ptcgo.txt	5/19/2017 8:38 PM	Text Document
card.ptcgo.txt	5/19/2017 8:38 PM	Text Document
deck1.ptcgo.txt	5/19/2017 8:37 PM	Text Document
deck2.ptcgo.txt	5/19/2017 8:38 PM	Text Document

Figure SEQ Figure * ARABIC 25File Browser

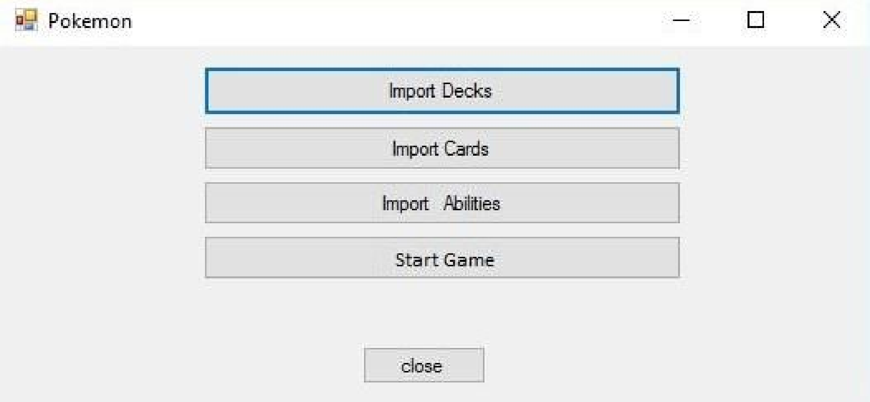
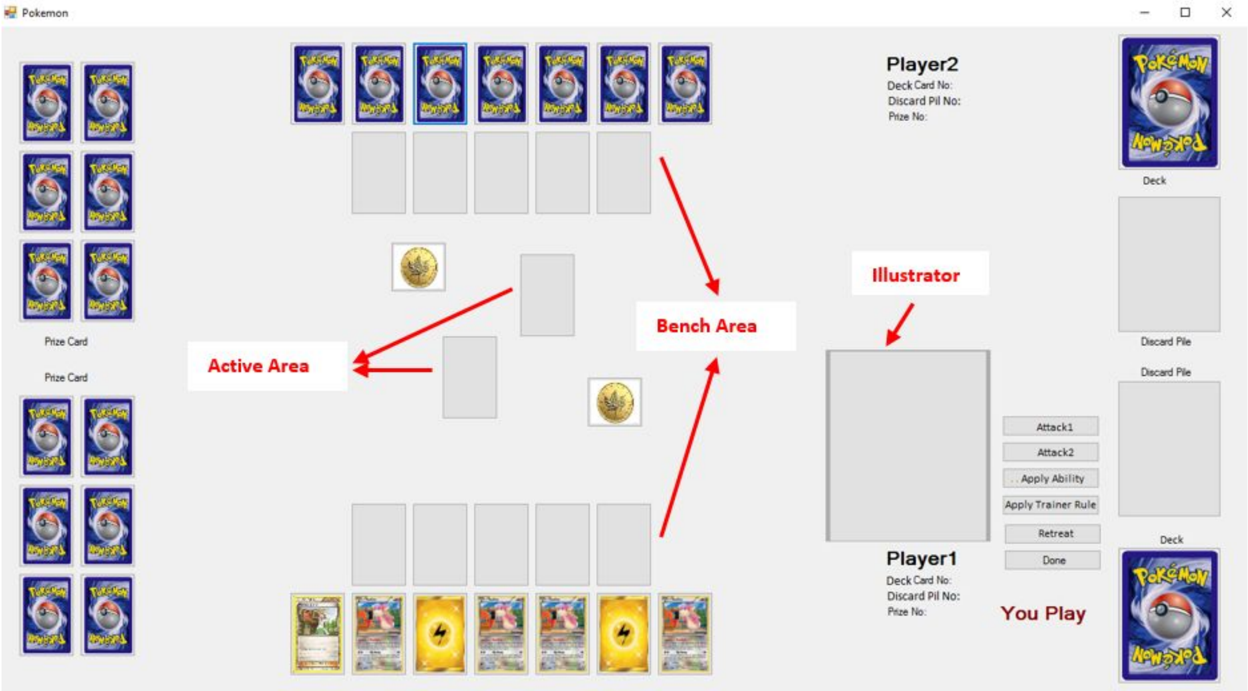


Figure SEQ Figure * ARABIC 246Start Game is enabled

3.2 Start Game

After Clicking on “Start Game” button, system shows the main board of the game, figure 27. System flips a coin and decide who starts first. Both Decks are shuffled, drew first seven cards and filled both Hands.



Then system draws six cards from and Decks and fills the Prize section.

3.3 Move Basic Pokémon to Bench or Active Area

3.4 Attach Energy to Pokémon

3.5 Evolve Basic Pokémon

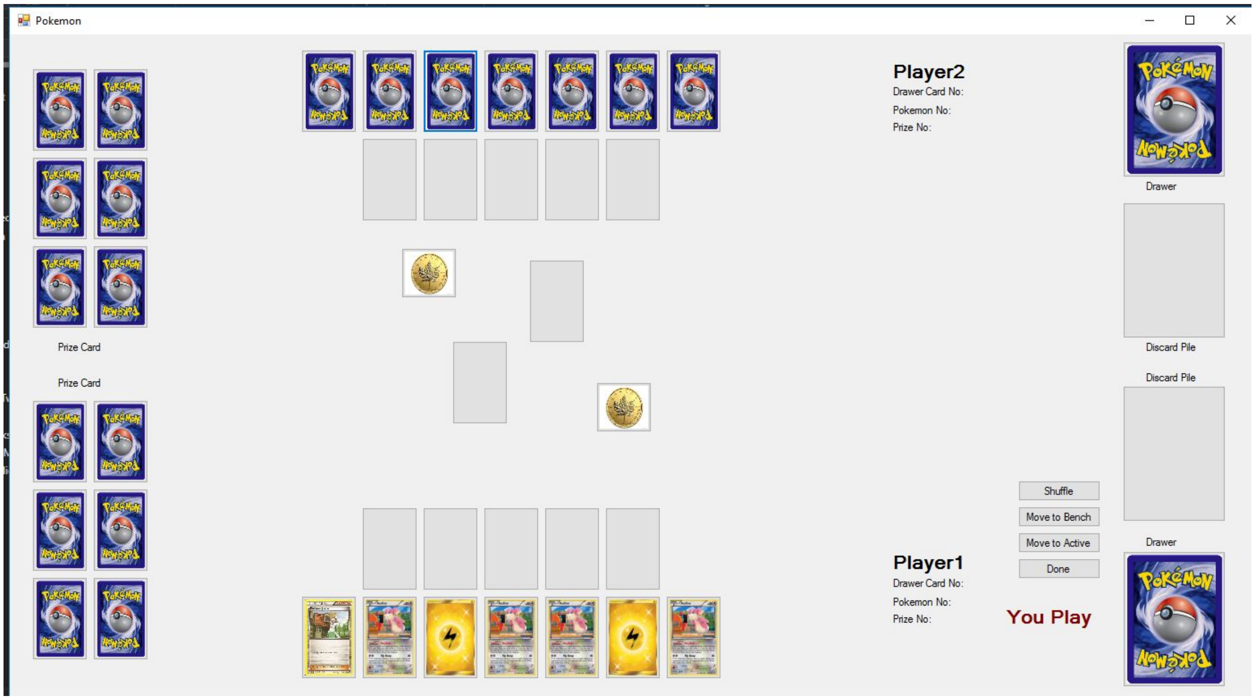
3.6 Select Item or Supporter Trainer

3.7 Use Ability

3.8 Retreat Pokémon

3.9 Attack Pokémon

3.10 Change Turn Manually


Pokemon
Player2
Drawer Card No:
Pokemon No:
Prize No:
Drawer
Discard Pile
Prize Card
Prize Card
Discard Pile
Shuffle
Move to Bench
Move to Active
Done
Drawer
Player1
Drawer Card No:
Pokemon No:
Prize No:
You Play

SECTION II: System Design Specification

4 Goals and Objectives

The purpose of the Software Design Specification (Section II) is to provide a high-level system architecture and design description of the PokémonGoBack software system to allow for software development to proceed with an understanding of what is to be built and how it is expected to build. It specifies the structure and design of some of the use cases discussed in the SRS (Section 1). It also displays some of the use cases that transformed into system sequence diagrams. The class diagrams show how the programming team would implement the specific module. This section consists of graphical documentation of the software design for the project including Domain Model, Sequence Diagrams, class diagrams, and other supporting requirement information. This section is divided into three major parts: operating environment, system architecture, and system behavior.

5 Operating Environment

1G processor.

Programming language: C#

IDE: Visual studio

We need .net 4.5 and Windows as OS.

6 System Architecture

6.1 Layered Architecture

In this project we used the concept of layer architecture. The Layered Architecture Style is focused around dividing software functionality into distinct layers that interact vertically. It is dependent on message passing between layers and clearly defined functional layers. Each layer can only send or

receive messages to the layer directly above or below it. Figure22 shows the layered architecture.

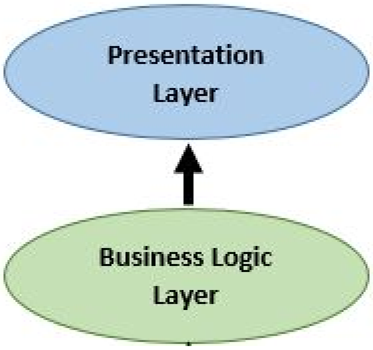
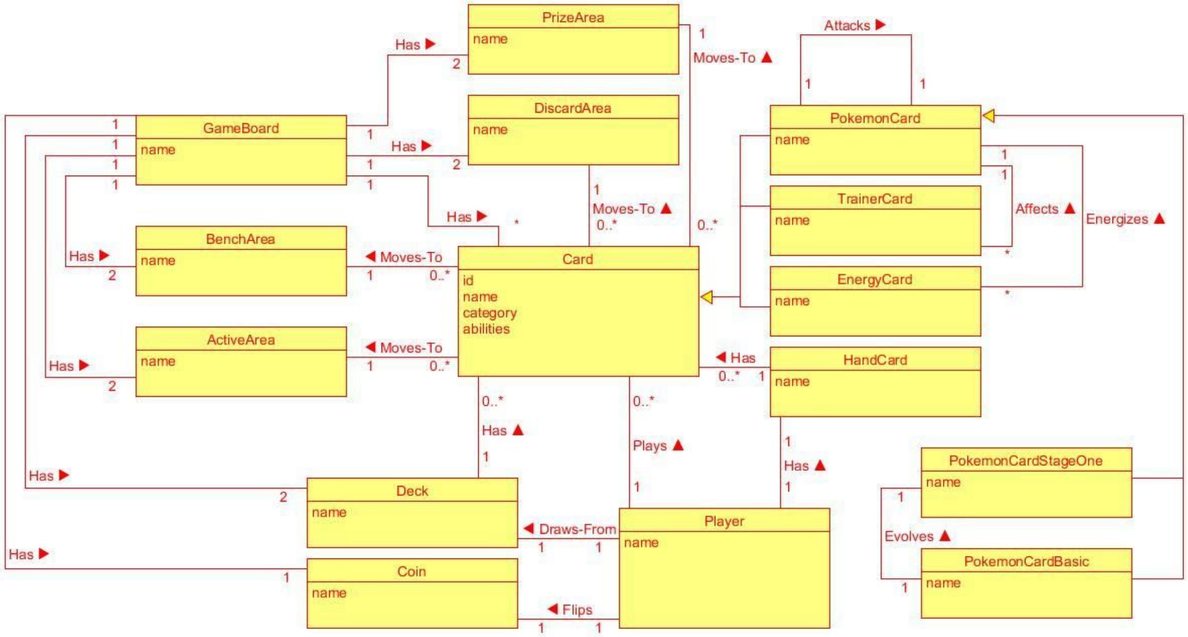


Figure SEQ Figure * ARABIC 28Layerd Architecture

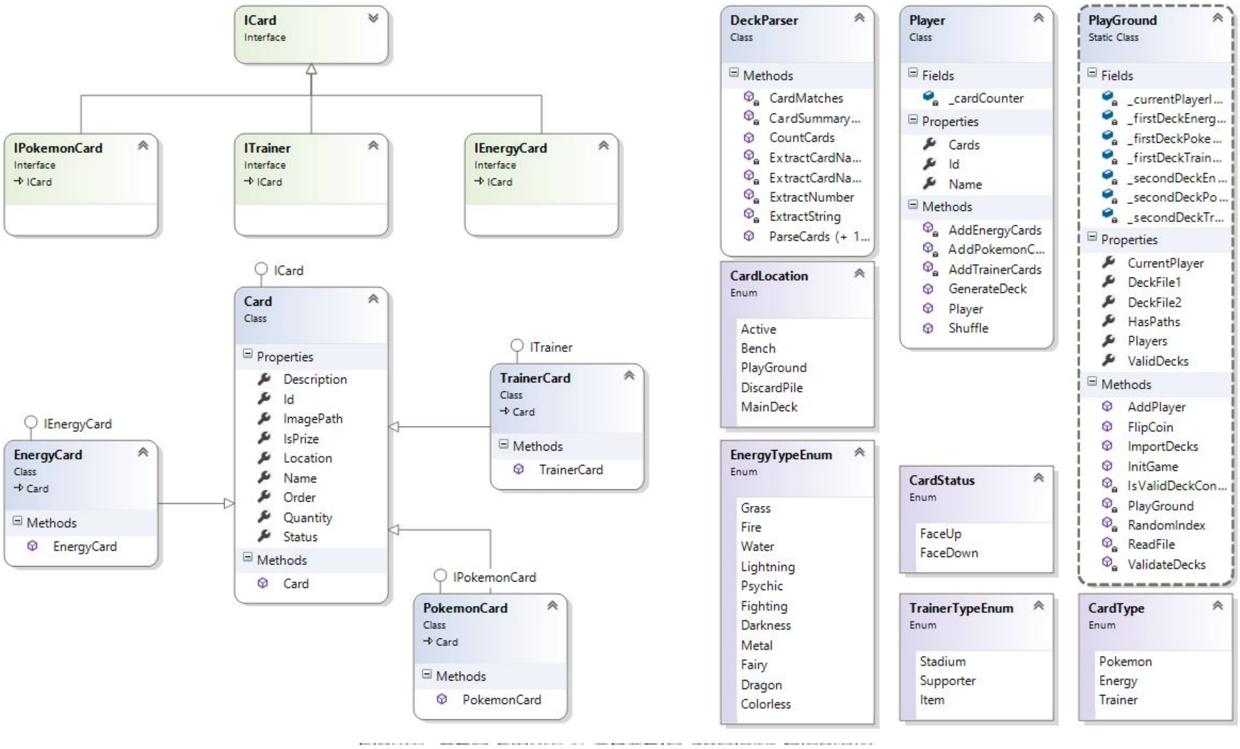
6.2 Domain Model



7

System Behavior Analysis

7.1 Class Diagrams



7.2 Sequence Diagrams

7.2.1 UC01-Import Files

7.2.2 UC02-Start Game

7.2.3 UC03 Move Basic Pokémon from Hand to Active or Bench

7.2.4 UC04-Attach Energy card to one Pokémon card

7.2.5 UC05-Evolve Basic Pokémon card

7.2.6 UC06-Use Ability

7.2.7 UC07-Retreat Active Pokémon

7.2.8 UC08-Select Item or Supporter Trainer Card

7.2.9 UC-09-Attack

7.2.10 UC10-Change Turn Manually

7.2.11 UC11-Damage Ability

Glossary and Index

To avoid confusion and prevent probable misunderstanding and definition over-rides, the following table lays out a set of terms and abbreviations, that are frequently used through-out this document, along with their definitions to act as a contract and provide a common ground between author of the document and its audience.

Table 6. Glossary

Terms	Use Cases	GUI Decision	Test
Agile			
Winer			
Opponent			
Turn			
SRS			
FR			
Draw			
Damage			
SSD			
Heal			
Stage-one Pokémon			
Basic Pokémon			
Attack			
Player			
System			
Activity Diagram			
UC			
Ability			
Deck			
Active Pokémon			
Benched Pokémon			
Hand			
Prize Section			
Discard Pile			
Shuffling			
Draw			
Deck Ability			
HP			
Search Ability			
Energy			
Retreat			
Trainer			
Stadium			
Item			
Evolve			

Supporter			
Destat Ability			
Swap Ability			
Redamage Ability			
Applystat Ability			
Deenergize Ability			
Reenergize Ability			
Cond Ability			
Add Ability			
Shuffle Ability			
Draw Ability			
Change game turn manually			
Done button			
Search Ability			
Use Ability			
Deck Ability			
Bench			

Reference