

---

# Pokémon Text-Audio Game

---



Report #1

[Repository](#)

[Repository Reports](#)

## **Team 3:**

[Rahil Sagarwala](#)

[Gary Ray](#)

[Luis Siavichay](#)

[Chad Mendenhall](#)

September 21, 2019



# Table of Contents

---

<b>Table of Contents</b>	2
<b>Project Management</b>	4
Individual Contributions.....	6
Rahil Sagarwala.....	7
Gary Ray.....	7
Chad Mendenhall.....	7
Luis Siavichay.....	8
Responsibility Matrix.....	8
Responsibility Allocation.....	9
<b>Customer Problem Statement</b>	10
Person 1.....	11
Person 2.....	12
Definitions.....	13
<b>System Requirements</b>	13
Enumerated Functional Requirements.....	14
Enumerated Nonfunctional Requirements.....	15
User Interface Requirements.....	15
<b>Functional Requirements Specification</b>	27
Stakeholders.....	28
Actors and Goals.....	28
Use Cases.....	29
Casual Description.....	30
Use Case Diagram.....	30
Traceability Matrix.....	<b>Error!</b>
<b>Bookmark not defined.</b>	
Fully Dressed Description.....	34
System Sequence Diagram.....	40
<b>User Interface Specification</b>	41
Preliminary Design.....	42
Navigation.....	42
Fight.....	45
UseItem.....	47
Catch.....	51
User Effort Estimation.....	54
Scenario 1.....	55

Scenario 2.....	55
<b>Domain Analysis</b>	57
Domain Model.....	57
SaveGame (UC-3).....	57
StartTradeRequest (UC-5).....	58
Concept Definition.....	58
Association Definitions.....	60
Attribute Definitions.....	61
Traceability Matrix.....	61
System Operation Contracts.....	63
Mathematical Model.....	63
<b>Project Size Estimation</b>	69
<b>Plan of Work</b>	69
<b>References</b>	71

# Project Management

---

Product: A Pokémon text audio game meant for people who have great nostalgia for the original games. This product is meant for the visually impaired as well. It is meant to be an easy game to pick up and play for anyone. This game will be in Java with hand-coded layout and will introduce complex features not seen in games similar to this one.

Responsibilities So Far:

**Rahil:**

- Worked on major aspects of report
- Worked on GUI aspect for code
- Started group off with coordination with Google Docs, Hangouts, and Github
- Created initial ideas for project and elaborated on ideas
- Shared resources with group (game scripts, soundtrack, mathematical information, YouTube tutorials, etc.)
- Accepted new members into the group

**Gary:**

- Helped with initial ideas on group discussion
- Collaborated on initial project ideas
- Worked on core aspects of report

Chad:

- Shared Google Hangouts link for video chat

Luis:

- Joined the group late and reviewed and analyzed the proposal and report
- Communicated with team on responsibilities for report and project
- Researched GUI code and shared links to support project
- Worked on design and layout of report

### Future Responsibilities:

We will be sharing all responsibilities at the moment.

Rahil:

- Work on reports
- Work on code

Gary:

- Work on reports
- Work on code

Chad:

- Work on reports
- Work on code

Luis:

- Work on reports
- Work on code

## Individual Contributions

---

### Rahil Sagarwala

1. Brainstormed ideas
  - a. 10+ initial ideas and broke it down to fewer ideas based upon group needs
2. Wrote 95% of proposal
3. Created Table of contents
4. Created Individual Contributions Page
5. Created customer problem statement
6. Created and sketched user interface requirements
7. Created Stakeholders list
8. Created Actors and Goals Table
9. Created Use Cases Casual Description Table
10. Created Traceability Matrix
11. Created User Interface Specification and added Preliminary Design sketches with explanations along with user effort estimation
12. Created “things to do” list for remaining objectives upon ~60-70% completion
13. Created Domain Model Concept Definitions Table, Associations Definitions Table, Attribute Definitions Table, Traceability Matrix, and domain model diagrams
14. Created Plan of Work
15. Created Responsibility Matrix
16. Created Fully Dressed Use Case Descriptions
17. Created Project Management Section
18. Added references

### Gary Ray

1. Wrote 5% of the proposal
2. Created the Functional Requirements Table
3. Created the Non-Functional Requirements Table
4. Created the Use Case Diagram
5. Created the System Sequence Diagram
6. Added references

### Chad Mendenhall

## Luis Siavichay

1. Late Introduction and Inclusion to team
2. Analyzing Ideas and Formats
3. Created Cover Page and Logo
4. Linked repositories and reports
5. Created Mathematical Model Tables and Graphs Diagrams
6. Tested the Formulas of the Mathematical Model for Accuracy in Excel
7. Enhanced Page Layout, Adjusted Tables, and Headings for Uniformity
8. Edited Report for Conversion to PDF



## Responsibility Matrix

		Team Member Name			
		Rahil Sagarwala	Gary Ray	Chad Mendenhall	Luis Siavichay
	Project Management (10 Points)	40%	30%	10%	20%
Responsibility Levels	Sec.1: Customer Problem Statement (9 points)	100%			
	Sec.2: System Requirements (6 Points)	15%	85%		
	Sec.3: Functional Requirements Specification (30 Points)	70%	30%		
	Sec.4: User Interface Specs (15 Points)	100%			
	Sec.5: Domain Analysis (25points)	75%			25%
	Sec.6: Plan of Work (5 points)	100%			

## Responsibility Allocation

# Customer Problem Statement

---

## Person 1

I have a friend who told me all about Pokémon and just how large the industry is. He told me that the games are brilliant and use the concept of animals as partners in an animated world. The one thing that I will always remember is that his childhood was consumed by his GameBoy Color 20 years ago as he played these role playing games. That childhood innocence is what saved his youth and put a smile on his face. Ever since, he has developed a passion for the games and the industry as a whole. Although it seemed to be the games that he went on and on about, he also enjoyed the show, cards, and any other merchandise he could find. He did tell me however, that everything else is secondary to the games. I was envious because I could never experience what he had. It was not that I wasn't interested. I was born with a rare condition and lost my sight at a very young age. I wish I could experience the joy that he feels to this day. I've heard descriptions of all the elements of this youthful passion, but I was never able to indulge.

In the 1990's when Pokémon became the next big thing, technology was not advanced enough to allow the minority population to enjoy entertainment. The original GameBoy itself is a device that runs on AA batteries and with a dim screen not comparable to what technology allows for today. Now today I am able to fully use a computer to browse the internet, read documents, or even write documents on my own. I have a keyboard that is advanced enough to allow me to type with ease. Even back when I was young, there were braille printable stickers that could be applied to keyboards. This was only the beginning of a solution to help those like me to enjoy the various aspects of life that others could enjoy. Unfortunately, there is no real solution to allowing me to use the GameBoy since its purpose is not meant to be changed. I've held a GameBoy before and the buttons were intuitive. There's an A button, B button, up, down, left, and right, start, select, and a volume slider. A direct solution to the issues that I face when it comes to this device, is the games itself. The device allows for ease of use, whereas the games do not. When my friend handed me his GameBoy with Pokémon Blue version, I was able to listen to the soundtrack and various sound elements within the game. I was not able to play however, because the games do not output any voice whatsoever. The only sensory elements found within the Pokémon games is sound without voice and the visual elements.

Since there is no direct solution to an ancient hardware, there is another solution I can think of, although more problems arise. I managed to install an emulator on my computer that can play Pokémon Blue. I was hoping that I could somehow use text to speech directly available on my computer in order to play. I was wrong and realized why when I understood that everything that goes on within the game, is in the game file itself. In essence, there is no way for me to play Pokémon Blue because of the way it was created. Of course, the device itself could have allowed for these capabilities, but one cannot blame the technological standards of that time.

One thing that I would like is a recreation of a Pokémon game such as Pokémon Blue version. I would want it to be on the computer and to output all the text as speech. I fell in love with the soundtrack as well and would like that to be implemented in each area of the game. It would be much easier for me if I could decide on which keys to use on the keyboard as well. I would like to be able to use my mouse for different parts of the game. The mouse is intuitive for me as it allows me to click on a few different areas in the palm of my hand. Of course it is understandable if this cannot be done for the entire game, however it

would be great to have that choice. One other problem that I have is that I'm unable to play video games with my friend. I would like to be able to play with him in some manner in this game. This would bring joy to others like me who want to feel a connection not just with the game, but fans of the franchise. The last thing that I would like is to be able to communicate with the creators of the game in some manner as this issue requires one with difficult problems to solve, to be able to voice his/her opinion on a game. Essentially ease of use can be made better through feedback from those with disabilities towards someone who cannot fathom all the problems that may arise.

## Person 2

I am a huge fan of the Pokémon series. In particular I love the games, especially Pokémon Blue, however there are some problems that I have. I've always wanted to play a text adventure game of this very game since it reminds me of my childhood but there doesn't exist one that is what I'm looking for. The soundtrack of these games is what gives me nostalgic feelings. That is something that I have not seen in a text based Pokémon game. Essentially, the problem that I have is that there exists Pokémon games that are either image, audio, and text, or just text. What I want is to play a game that introduces a new realm of games that merges the idea behind a text based adventure and an audio game. It is not as simple as me playing one of the existing text based games out there along with listening to soundtracks on YouTube. This requires too much work especially since I would have to switch the music every few minutes and many more times, especially if I am going through frequent battles.

Another issue that I have is that current text based games do not have multiplayer support. I would like to reminisce of the time when I was a child and playing with my friends using the link cable. This connection between people is what made Pokémon such an amazing series. It allowed people to compete with each other but also and most importantly, it allowed people to trade Pokémon with each other. Something that I would like to see that has not been done well in a text based adventure Pokémon game, is the concept of evolution. Some Pokémon can only evolve after trading with a friend and I believe introducing this concept will make this game authentic, just like the real thing in terms of mechanics. It just isn't a Pokémon game if trading isn't beneficial in some way.

When I am playing a game, I like to see a visually nice display. Nothing over the top, but clean. Existing text based Pokémon games are messy and are not eye catching. Perhaps to capture my nostalgia, one could have a background of the actual GameBoy device with the screen of the GameBoy playing the game. There's something to be said about a visually clean game, and those are the games that usually draw me in.

Usually at night when I'm in bed, I like to listen to music in the dark. I have a computer with a lit up keyboard and it would be amazing if I could use this keyboard to play the game without having to look at the screen. I would like the game to read the text to me like a novel, except I am controlling what goes on.

The final issue I have with existing games in general is the fact that save files are not implemented with ease of use. The original games and even to this day, allow for only one save file. Most text based games that I have seen have this same problem as well. What I would like is to be able to decide which save file I would like to play at the beginning of the game and to be able to rename these saves as I please. It would be great if there was ability to upload saves automatically to some cloud based storage such as google drive, since that is what I use. This way, there would be no issue with losing progress on a game if there is a malfunction on my computer.

## Definitions

**GameBoy Color:** A device released in 1998, first in Japan and then worldwide a few months later. This device can play GBC (gameboy color) games.

**Soundtrack:** The music within the game that, constantly playing and changing.

**Pokémon:** Animal like creatures with attack moves, stats, evolutions, levels, etc.

**Trade:** To exchange a Pokémon with a friend.

**Battle:** To use one's Pokémon to fight another Pokémon.

**Link Cable:** A cable for the Gameboy Color to connect two devices and allow multiplayer support (Max 2 players).

**Multiplayer:** When 2 or more people can play the same game either on a different device on the same device (in this case, different device).

**Evolution:** When a Pokémon reaches a certain level, it evolves into a more powerful Pokémon.

**Text Based Game:** A game with only text on the display

**Role Playing Game:** A game meant to put the player in the character of the game's shoes.

**Cloud:** Storage using databases and servers such as Google Drive, Onedrive, DropBox, etc.

**Save File:** Saving a game creates a file that allows the user to pick up from where he/she left off.

## System Requirements

---

Enumerated Functional Requirements		
Req-x	Priority Weight	Req. Description
Req-1	3	User will be able to trade Pokémon with friends through maintained database
Req-2	1	The game will be able to produce a rating survey
Req-3	1	The game will have a soundtrack provided
Req-4	3	The game will provide numerous accessibility options
Req-5	2	User will be able to choose the save file he/she will be using
Req-6	2	Users will be able to save game file
Req-7	4	User will have control over deciding the character's navigation through location options
Req-8	3	User will be able to start a new game
Req-9	5	Pokémon will be able to fight other Pokémon
Req-10	5	User will be able to catch wild Pokémon
Req-11	3	User will be able to heal Pokémon at Pokémon center
Req-12	3	User can withdraw/Deposit Pokémon/items from PC
Req-13	4	NPC Trainers/Wild Pokémon Initiate Battle
Req-14	5	User will be able to use items in menu or in battle

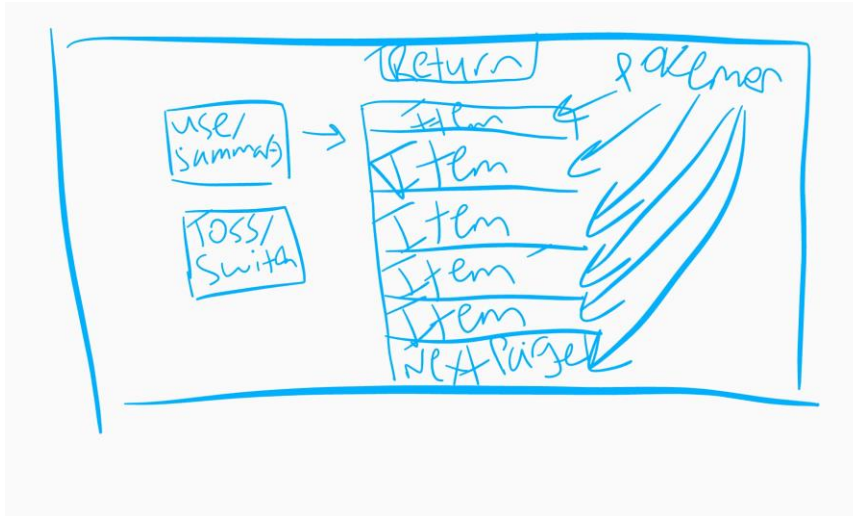
## Enumerated Nonfunctional Requirements

Req-x	Req. Description
Req-15	Trade will have to be accepted by both users in order to have it approved.
Req-16	Rating option will be displayed at the start screen. Users will have the option to accept or decline.
Req-17	User will have ON/OFF setting for soundtrack music in title screen options
Req-18	Text to Speech will be an accessibility option for visually impaired users.
Req-19	If gameplay requires multiple characters, additional options will be added to dictate which character is in use at that moment.
Req-20	Gameplay options like attack or retreat will be supplied to the user (combat system)
Req-21	Game will have a myriad amount of location (cities/towns)
Req-22	Game will contain storage Pokémon array for all Pokémon in PC.
Req-23	Locations will have wild Pokémon through use of random numbers

## User Interface Requirements

**Note:** Disregard top menu buttons from the following screens:  
Game Dialogue, Pokémon Battle, Trainer Battle Initial Screen, and Win/Lose

### Bag and Party



#### Bag:

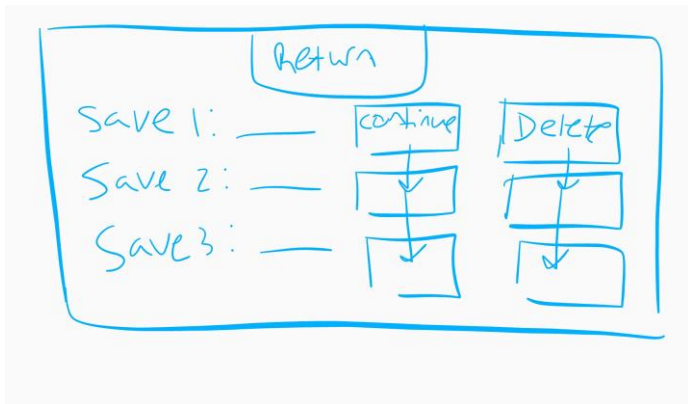
- From Items button in Menu screen/Pokémon Battle Screen Bag Button
- Return Button goes to Menu screen/Return button goes back to Pokémon Battle screen
- Clicking on Item button allows 2 new options to appear: Use and Toss
  - Use button applies effect of item and decreases amount, if amount less than one, remove from bag
  - Toss button decreases item by one, if less than 1, remove from back

#### Party:

- From Party Button in Menu Screen/Pokémon button in Pokémon Battle Screen
- Return Button goes back to Menu Screen/Pokémon Battle Screen
- Clicking on a Pokémon button allows 2 new buttons to appear: summary and switch
  - Summary button goes to Pokémon Menu Battle Summary Screen
  - Switch button Shifts every Pokémon in array of 6 down by 1

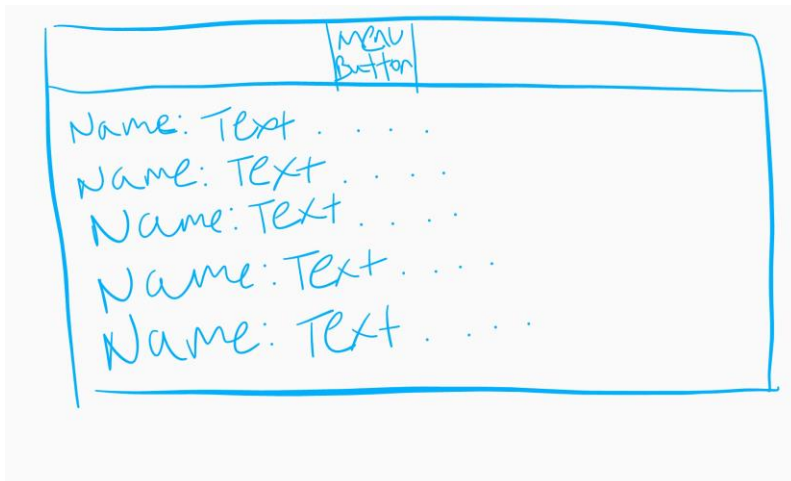


## Continue Game

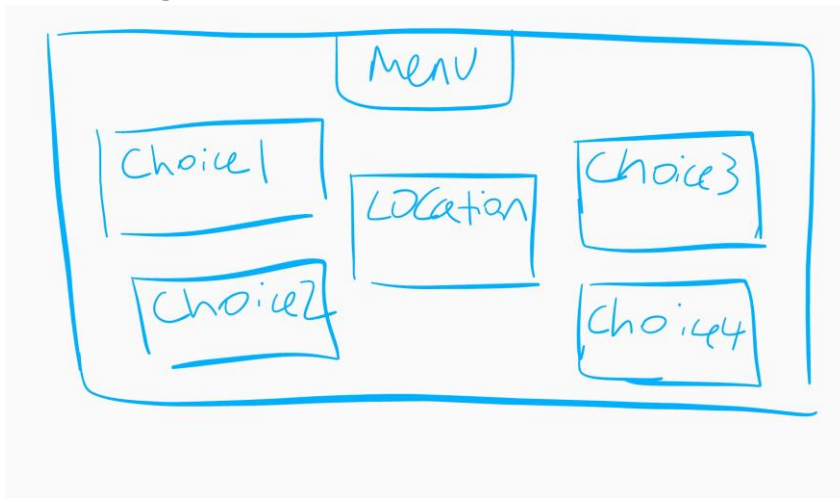


- From Start Screen Continue button
- Returns to Start Screen
- Save File names are shown, max of 3
- 2 buttons per save file
  - Continue: Picks up from where left off
  - Delete: Removes link of save file and sets name to empty

## Game Dialogue

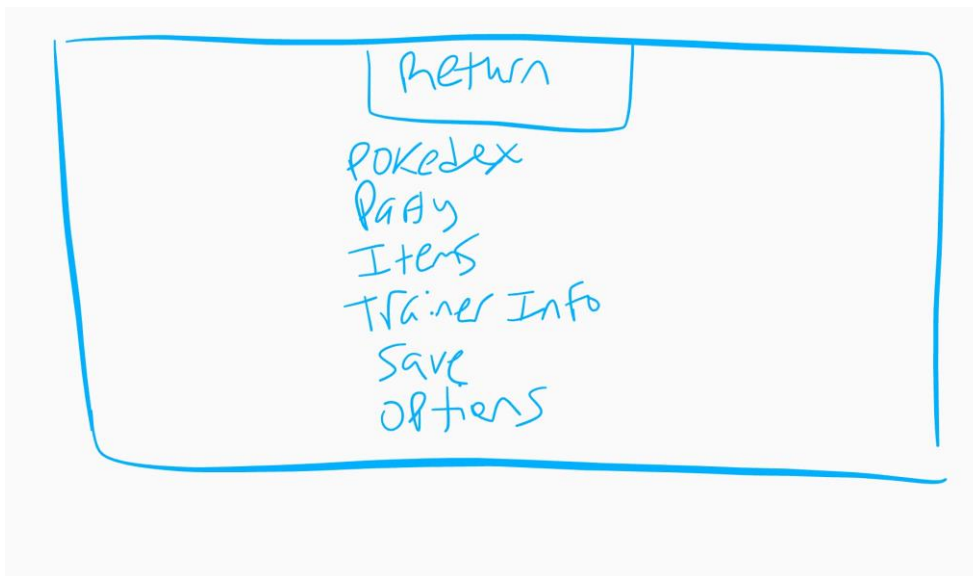


## Game Navigation



- Location text in center
- 4 buttons with choices of locations

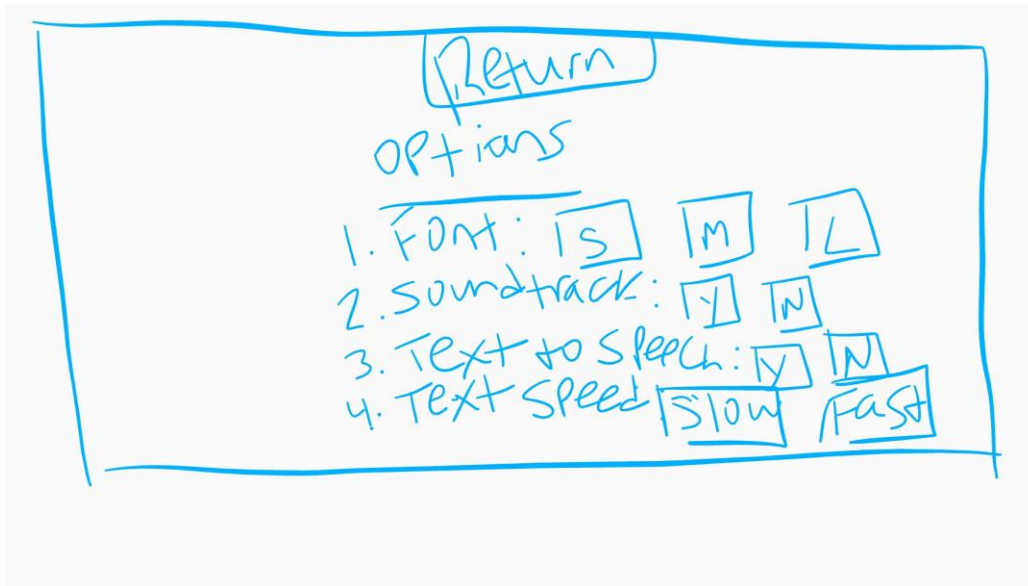
## Menu



- From Game Dialogue/Game Navigation
- Return to Game Dialogue/Game Navigation
- Pokedex Button: Go to Pokedex Screen
- Party Button: Go to Pokémon summary screen

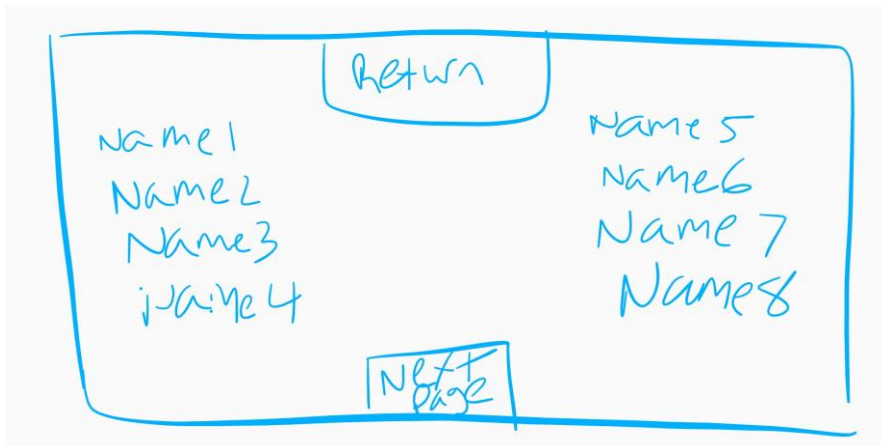
- Items Button: Go to bag screen
- Trainer Info Button: Go to trainer info screen
- Save Button: Go to save screen
- Options Button: Go to options screen

## Options



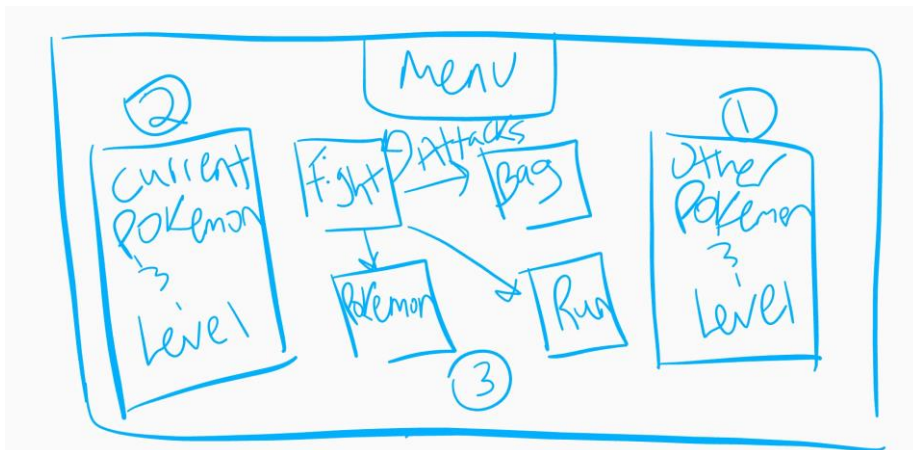
- From options button in start screen
- Return to start screen
- Each option has text followed by buttons
  - Font: Change font to small, medium, large
  - Soundtrack: Turn on/off the music
  - Text to Speech: Turn on/off text to speech
  - Text Speed: Instant text on game dialogue/slow text on game dialogue

## Pokedex



- From Pokedex button in menu
- Returns to menu
- Each Pokémon's name is a button that can be clicked to go to Pokémon Pokedex Entry
- Next Page button scrolls through array and changes links/names on buttons

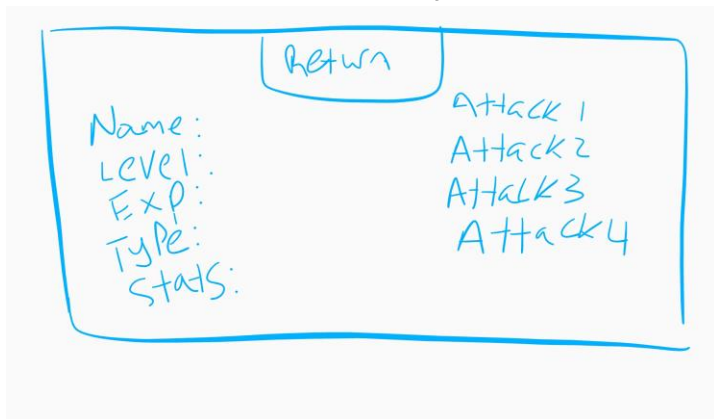
## Pokémon Battle



- First show other Pokémon, then current Pokémon, then 4 middle buttons
- Fight Button changes the 4 middle buttons to attacks
- Pokémon button goes to Pokémon Summary
- Bag Button goes to Items
- Run returns to Game Navigation if and only if this is not a trainer battle

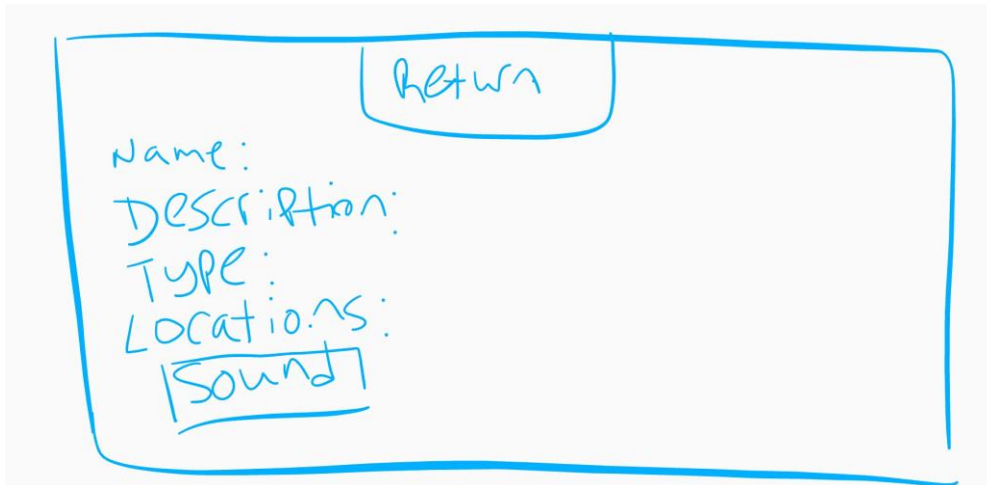
- If a Pokémon's HP is less than 0, go to win/lose screen

### Pokémon menu battle summary



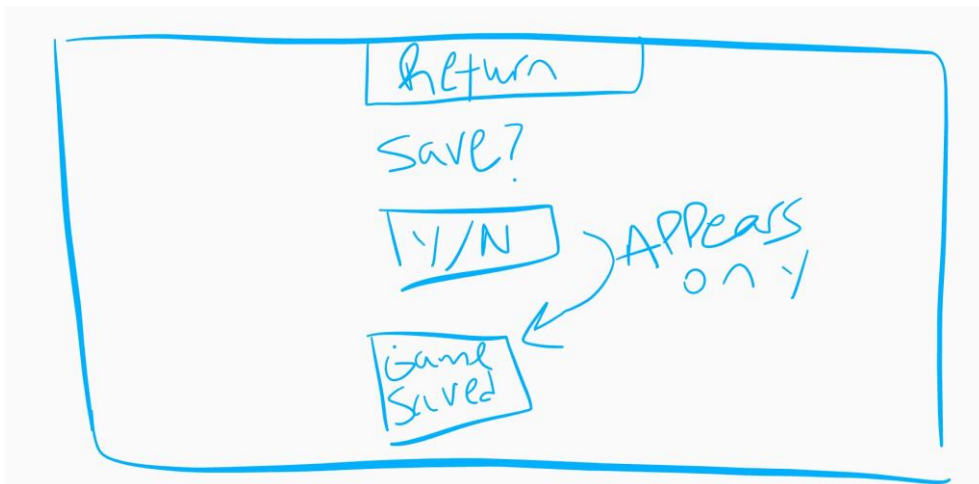
- From summary button on Party
- Returns to Party screen
- Everything on this screen is text

## Pokémon Pokedex entry



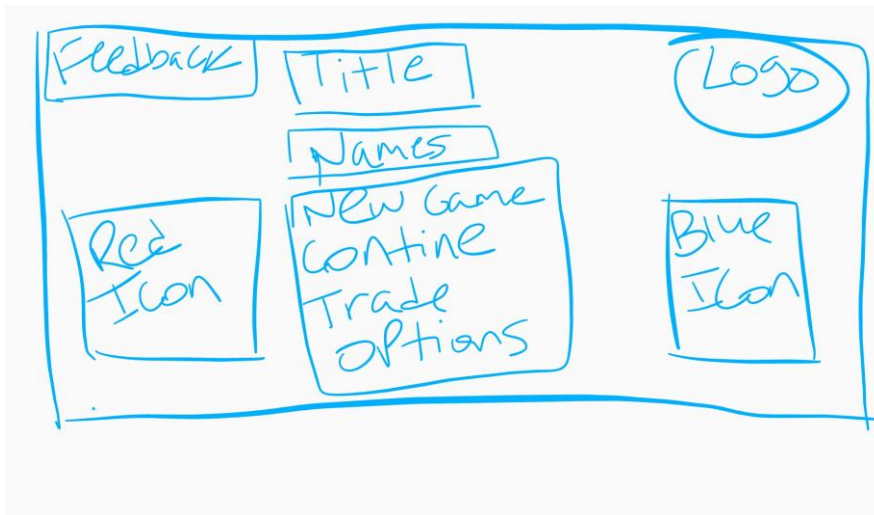
- From Pokedex button in menu
- Return to menu
- Only button is sound which plays the sound of the Pokémon

## Save Game



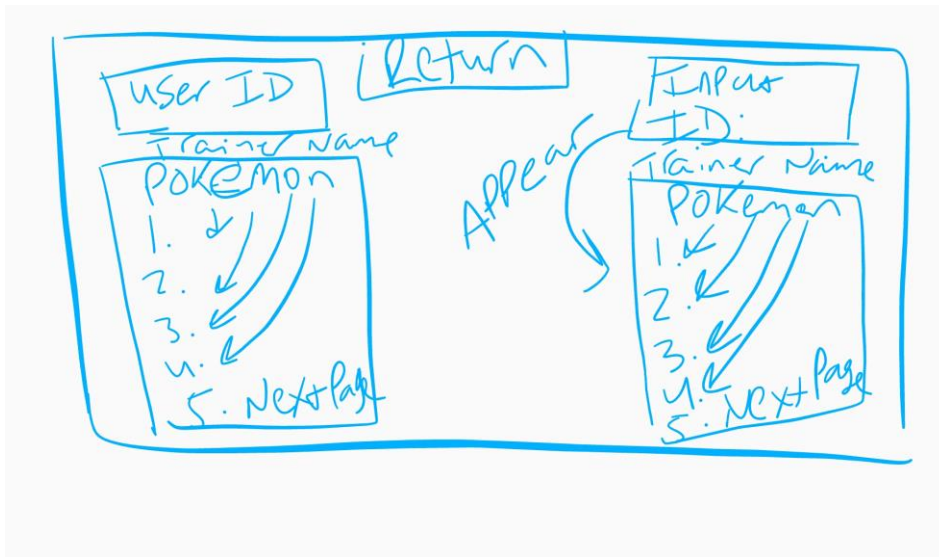
- From Save Game button in menu
- Return to menu
- “Save?” Text with 2 buttons: yes and no
  - Yes: Change text to “game saved” and return to menu
  - No: Return to menu

## Start Screen



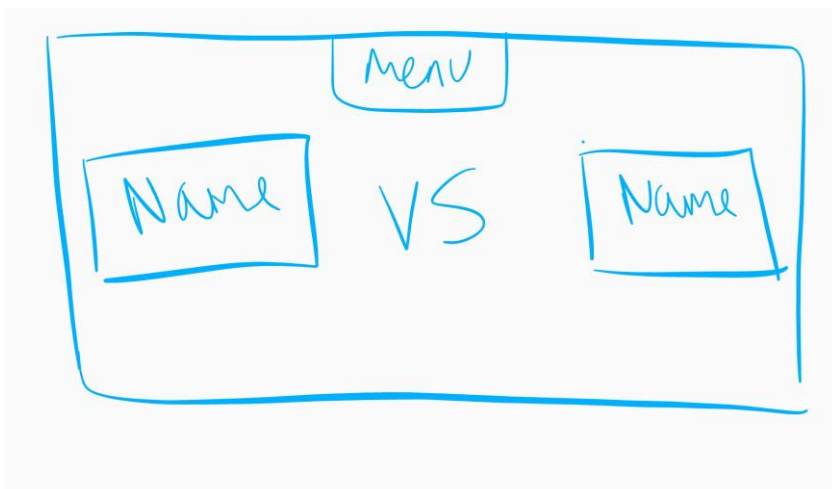
- Feedback button opens a link in a browser to a feedback page
- New Game button starts the game from fresh
- Continue button goes to continue screen
- Trade button goes to trade screen
- Options button goes to options screen

## Trade



- Text field to input ID, searches database, finds ID, searches save file, and outputs Trainer name with Pokémon info
- Each Pokémon represented as a button
  - Clicking on one of your own and one of other trainer's creates new text and buttons in middle: "send trade request?", "yes", "no". Either option returns to start screen.

## Trainer Battle Initial screen





## Trainer Info

Return

Name: ID:

# Pokemon Have:

# Gym Badges have:

Amount of money:

- From Trainer Info button in menu
- Returns to Menu

## Win Lose

Menu

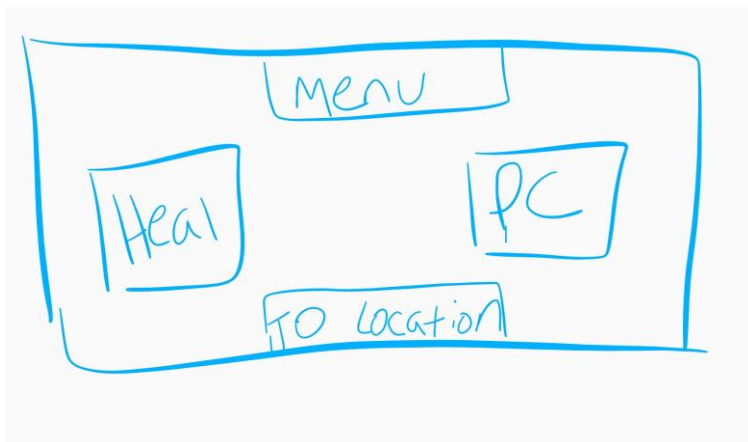
Name Win/Lose

(Disappear if Lose)

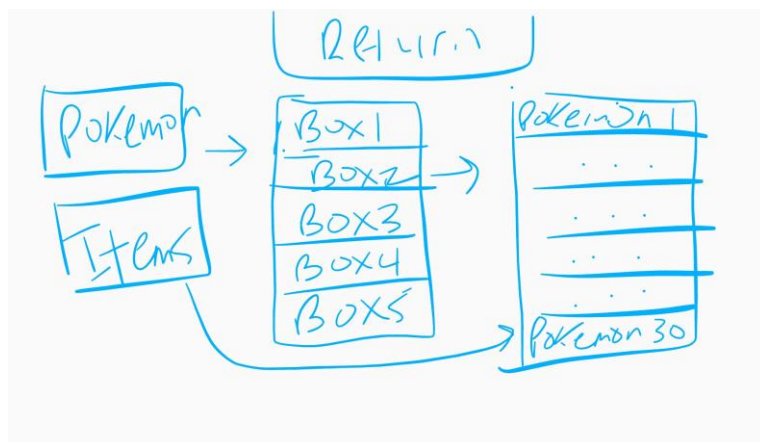
Name Win/Lose

- Whichever one loses, play sound and make their name disappears

## Pokémon Center

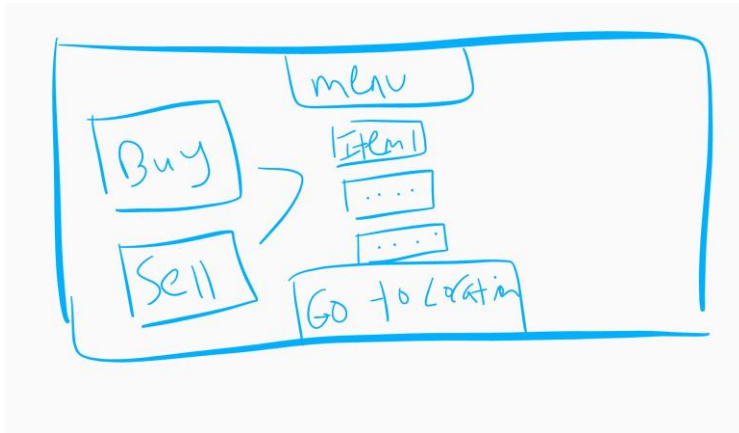


## PC



- Each Box for Pokémon can hold up to 30 Pokémon in an array
- Items goes directly to item selection

## Market



- Buy and Sell lead to list of items

# Functional Requirements Specification

---

## Stakeholders

- Person with visual disability (direct)
- Person with close ties to Pokémon games (direct)
- Person learning to read/speak English (For example, children or international students) (potential)

## Actors and Goals

Actor	Type	Goals	Use Case Name
Computer Player	Initiating	To start up game	StartGame (UC-1)
Computer Player	Initiating	To navigate through game locations/menu/options	NavigateGame (UC-2)
Computer Player	Initiating	To save game	SaveGame (UC-3), UC-2
Computer Player	Initiating	To load game	LoadGame (UC-4), UC-2
Computer Player	Initiating	To initiate trade request	StartTradeRequest (UC-5), UC-2
Computer Player	Initiating	To initiate Item use	UseItem (UC-6), UC-2
Computer Player	Initiating	To submit feedback	Survey (UC-7), UC-2
Computer Player	Initiating	To catch Pokémon	Catch (UC-9), UC-2

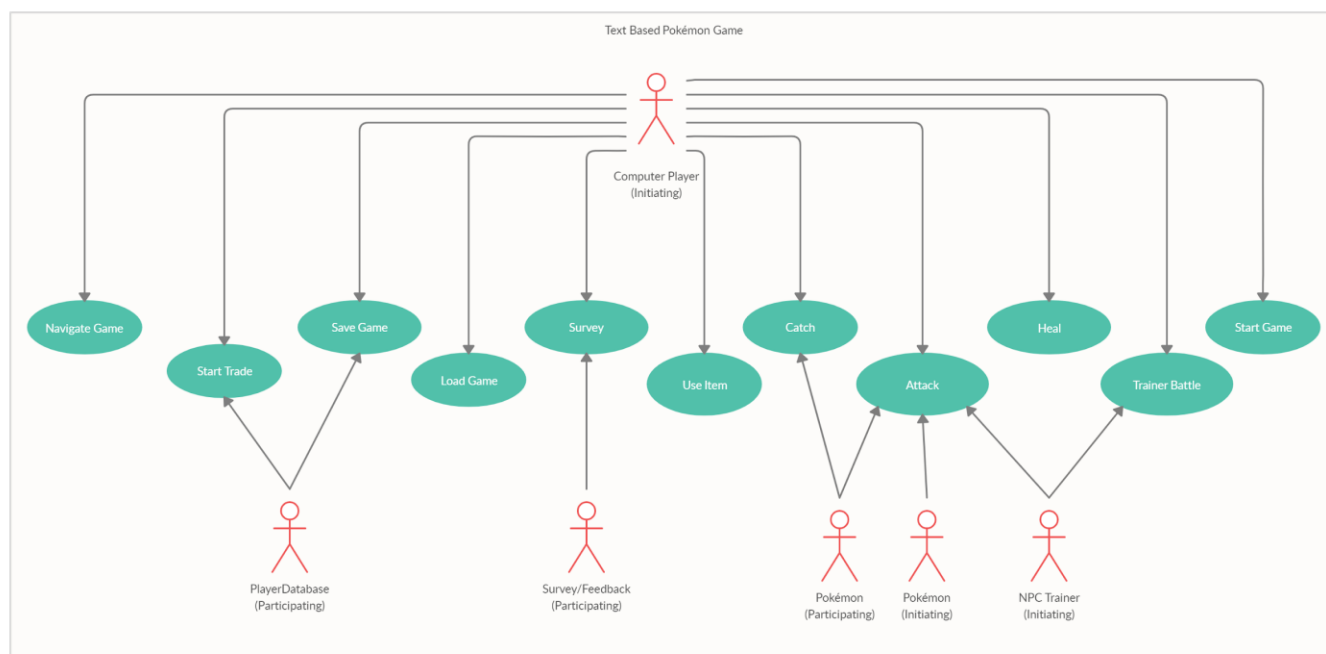
Computer Player	Initiating	To heal Pokémon	Heal (UC-10), UC-2
Pokémon	Participating/ Initiating	To attack other Pokémon and win/lose	Attack (UC-8), UC-2
Pokémon	Participating	To be caught	(UC-9)
Trainer NPC	Initiating	To battle Computer Player	TrainerBattle (UC-11)
Trainer NPC	Initiating	To use his/her Pokémon to attack computer player's Pokémon	UC-8
Player Database	Participating	To retrieve and mark elements of player from save file and player ID	UC-3, UC-5
Survey/Feedback Site	Participating	To allow computer player to fill out a feedback/survey form and then submit	UC-7

## Use Cases

### Casual Description

Use Case	Description	Requirements
StartGame (UC-1)	Starts a new game from the beginning	Req-8
NavigateGame (UC-2)	Change locations based upon 4 given options, main game settings based upon a few options for each, game menu options based upon 6 separate options, etc.	Req-7, Req-4
SaveGame (UC-3)	Overwrite current save game file	Req-6
LoadGame (UC-4)	Picks up from where game left off last time upon exit	Req-5
StartTradeRequest (UC-5)	Send trade request to database, which then goes to other user	Req-1
UseItem (UC-6)	Use an item depending on its effects, -1 count for each item use	Req-14
Survey (UC-7)	External link to survey page	Req-2
Attack (UC-8)	Pick between 4 moves on existing Pokémon to fight other Pokémon	Req-9
Catch (UC-9)	Catching a Pokémon that is eligible to be caught (not another trainer's) will be sent to party. If party full (6 slots occupied), send to storage.	Req-7, Req-10
Heal (UC-10)	Heal all Pokémon in current party to max HP	Req-11
TrainerBattle (UC-11)	NPC Trainers start an inescapable battle sequence	Req-13

## Use Case Diagram



## Traceability Matrix

Req't	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11
REQ1	3					X						
REQ2	1					X		X				
REQ3	1											
REQ4	3		X									
REQ5	2				X							
REQ6	2			X								
REQ7	4		X							X		
REQ8	3	X										
REQ9	5								X			
REQ10	5									X		
REQ11	3										X	
REQ12	3											
REQ13	4											X
REQ14	5						X			X		
Max PW		3	4	2	2	3	5	1	5	5	3	4
Total PW		3	7	2	2	4	5	1	5	14	3	4



REQ1: Trade Pokémon through database

REQ2: Produce a rating survey

REQ3: Soundtrack provided

REQ4: Numerous accessibility options

REQ5: Choose the save file

REQ6: Save game file

REQ7: Control over navigation

REQ8: Start new Game

REQ9: Pokémon fight other Pokémon

REQ10: Catch wild Pokémon

REQ11: Heal Pokémon at Pokémon center

REQ12: Deposit/Withdraw items/Pokémon at PC

REQ13: NPC/wild Pokémon initiates battle

REQ14: Use Item in menu/battle

UC1: StartGame

UC2: NavigateGame

UC3: SaveGame

UC4: LoadGame

UC5: StartTradeRequest

UC6: UseItem

UC7: Survey

UC8: Attack

UC9: Catch

UC10: Heal

UC11: TrainerBattle

## Fully Dressed Description

Use Case UC-8 Attack	
Related Requirements	Req 9
Initiating Actor:	Computer Player, Pokémon
Actor's Goal:	To defeat other Pokémon by using his/her Pokémon's attacks
Participating Actor:	Pokémon
Preconditions	<ol style="list-style-type: none"> <li>1. Computer Player's Pokémon must be alive</li> <li>2. Attack must be a valid defined attack</li> <li>3. Pokémon must be at or below a certain level to attack depending on badge count, otherwise it will not obey</li> </ol>
Postconditions	<ol style="list-style-type: none"> <li>1. System must look for attack power</li> <li>2. System must look for attack type with strengths/weaknesses</li> <li>3. System must look for attack effect</li> <li>4. System must apply attack to opposing Pokémon's HP</li> </ol>
Flow of Events for Main Success Scenario	
1. →	<b>Computer Player</b> Initiates an attack out of 4 choices
2. →	<b>Pokémon</b> object uses its attack

3. ←	<b>System</b> determines current Pokémon's power, attack power, strengths, weakness, and all over the above for opposing Pokémon
4. →	<b>Opposing Pokémon</b> receives damage
<b>Flow of Events for Extensions (Alternate Scenarios)</b>	
1. →	<b>Computer Player</b> Initiates an attack out of 4 choices
2. →	<b>Pokémon</b> object uses its attack
3. ←	<b>System</b> fails to check attributes
1. →	<b>Computer Player</b> Initiates an attack out of 4 choices
2. →	<b>Opposing Pokémon</b> attacks first
3. →	<b>Pokémon</b> receives damage and faints
4. →	<b>Pokémon</b> object uses its attack
1. →	<b>Computer Player</b> Initiates an attack out of 4 choices
2. →	<b>Pokémon</b> object uses its attack
3. ←	<b>System</b> fails to check for level
4. →	<b>Opposing Pokémon</b> receives damage

Use Case UC-6 UseItem	
Related Requirements	Req 14
Initiating Actor:	Computer Player, NPC Trainer
Actor's Goal:	To use an item inside or outside of battle to apply effect
Participating Actor:	
Preconditions	<ol style="list-style-type: none"> <li>1. Item must be applicable to situation</li> <li>2. Item must be available in inventory/bag</li> <li>3. Item cannot be a key item in battle</li> </ol>
Postconditions	<ol style="list-style-type: none"> <li>1. Item array must be reduced by 1</li> <li>2. Item must apply effect</li> </ol>
Flow of Events for Main Success Scenario	
1. →	<b>Computer Player</b> initiates item use (potion) inside battle
2. ←	<b>System</b> determines item's effect
3. ←	<b>System</b> reduces item array by 1
4. →	<b>NPC Trainer</b> initiates item use (potion) inside battle

5. ←	<b>System</b> reduces item array by 1
<b>Flow of Events for Extensions (Alternate Scenarios)</b>	
1. →	<b>Computer Player</b> initiates key item use (fishing rod) inside battle
2. ←	<b>System</b> determines item's effect, displays fishing text
3. ←	<b>System</b> reduces item array by 1
1. →	<b>Computer Player</b> initiates item use (potion) inside battle
2. ←	<b>System</b> determines item's effect
3. ←	<b>System</b> does not reduce item array by 1

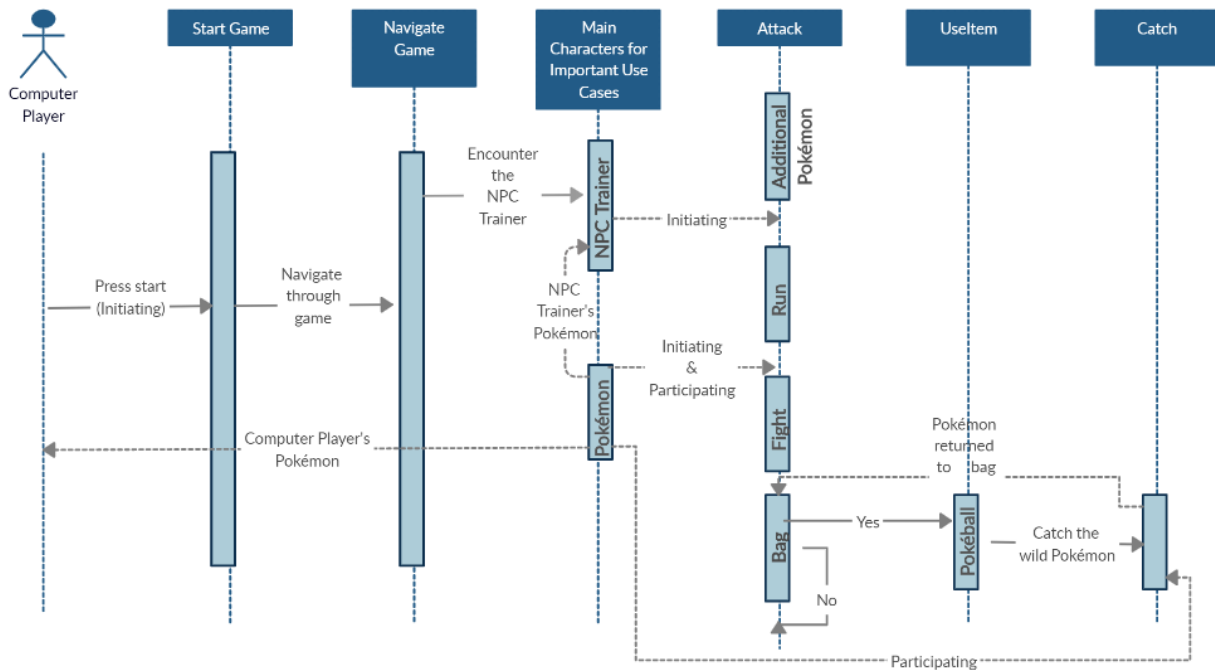
<b>Use Case UC-9 Catch</b>	
Related Requirements	Req 7, Req 10, Req 14
Initiating Actor:	Computer Player
Actor's Goal:	To add a wild Pokémon to party/box
Participating Actor:	Pokémon
Preconditions	1. Pokémon cannot be an NPC Trainer's Pokémon 2. Pokéball item must be used 3. Can only be done in battle

Postconditions	<ol style="list-style-type: none"> <li>1. The same exact Pokémon must be added to box/party</li> <li>2. If party not full, add Pokémon to party</li> <li>3. If party full, search for box</li> <li>4. If box full, create another box and add Pokémon to it</li> </ol>
<b>Flow of Events for Main Success Scenario</b>	
1. →	<b>Computer Player</b> initiates item use (Pokéball) inside battle
2. ←	<b>System</b> determines item's effect
3. ←	<b>System</b> decides Pokémon is caught
4. ←	<b>System</b> stores Pokémon into correct location
5. ←	<b>System</b> decides that battle has ended and returns to previous location
<b>Flow of Events for Extensions (Alternate Scenarios)</b>	
1. →	<b>Computer Player</b> initiates item use (Pokéball) inside battle
2. ←	<b>System</b> determines item's effect
3. ←	<b>System</b> decides Pokémon is not caught (random)
1. →	<b>Computer Player</b> initiates item use (Pokéball) inside battle on NPC Trainer's Pokémon
2. ←	<b>System</b> determines item's effect

3. ←	<b>System</b> decides Pokémon is caught
4. ←	<b>System</b> stores Pokémon into correct location
5. ←	<b>System</b> decides that battle has ended and returns to previous location
1. →	<b>Computer Player</b> initiates item use (Pokéball) inside battle
2. ←	<b>System</b> determines item's effect
3. ←	<b>System</b> decides Pokémon is caught
4. ←	<b>System</b> cannot store Pokémon into correct location
5. ←	<b>System</b> decides that battle has ended and returns to previous location, Pokémon not caught



## System Sequence Diagram



This system sequence diagram covers 3 of our most important use cases. The attack, UseItem, and Catch use cases are represented in the diagram.

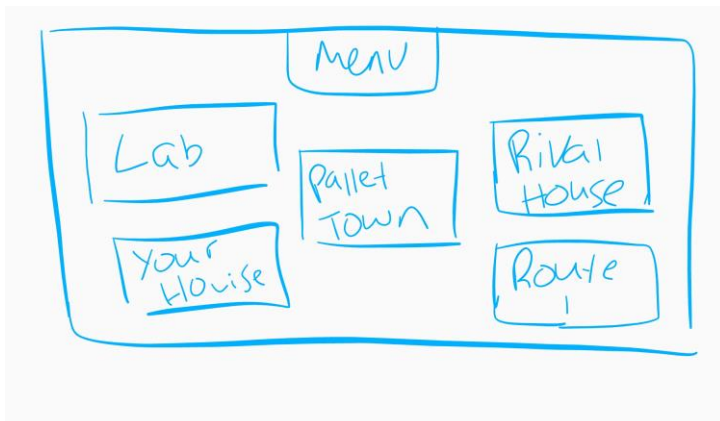
# User Interface Specification

---

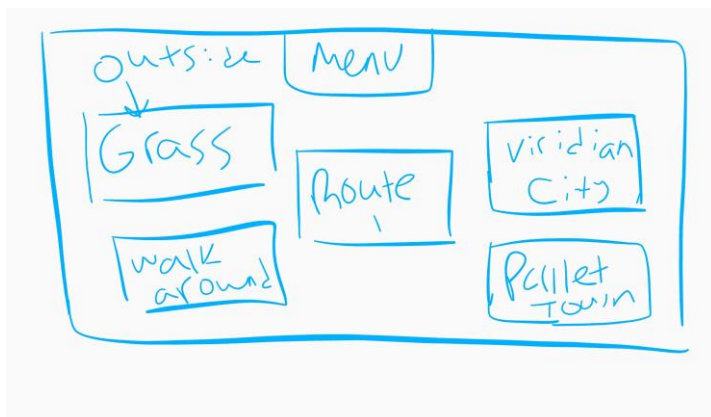
## Preliminary Design

The most important function of the game is the battle system, thus it is important to discuss the mechanics of battling alongside how battles are initiated. In many circumstances, the player does not have a choice to initiate battles. Only specific battles does the player have a choice. Otherwise, these are random in nature depending on if a Pokémon is randomly encountered in grass area (other areas included) or if a built in NPC trainer spots the player randomly. Although Navigation is not a top priority in terms of weight, it will be demonstrated here.

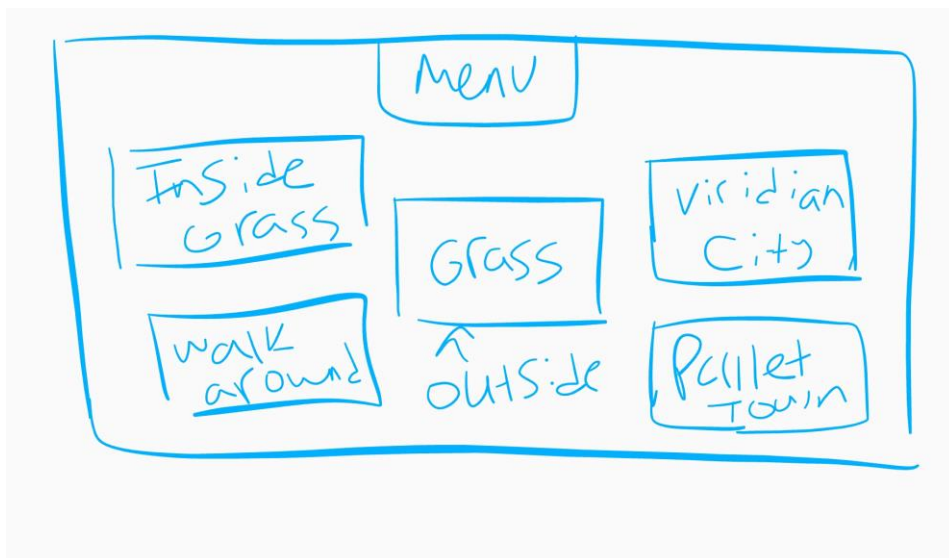
## Navigation



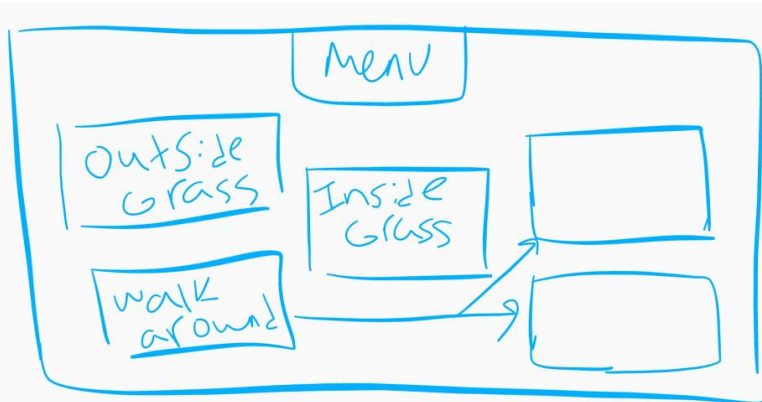
This is most likely the starting options after the initial dialogue when user starts the game. The goal for this example, is to head to an area with wild and random Pokémon encounters, which will be in route 1.



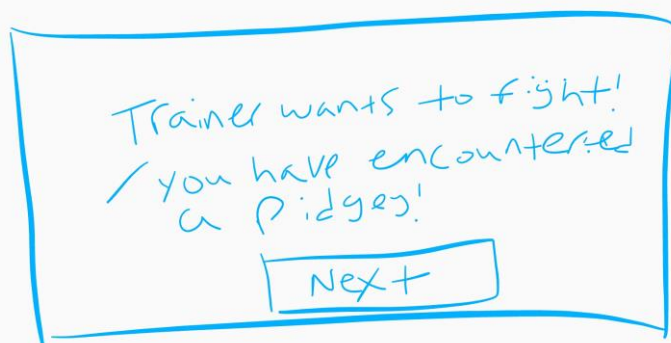
The goal is to encounter a random wild Pokémon inside route 1 is inside the grass. Walking around Route 1 may lead to a trainer encounter and thus a trainer battle.



Once outside of the grass, the goal is to go inside the grass.

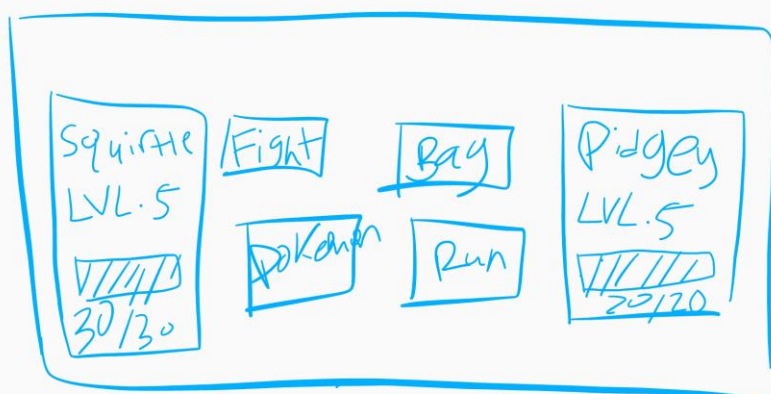


The only options inside grass is to walk around or return to outside grass. Walking around inside grass will inevitably lead to a random Pokémon encounter.

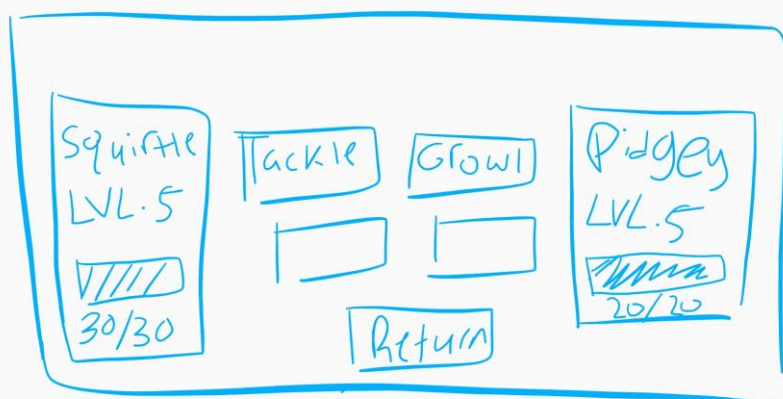


In this case, a Pidgy has been encountered

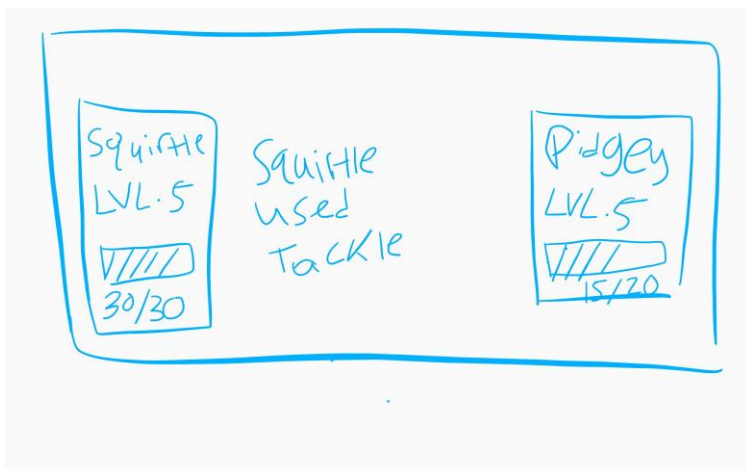
## Fight



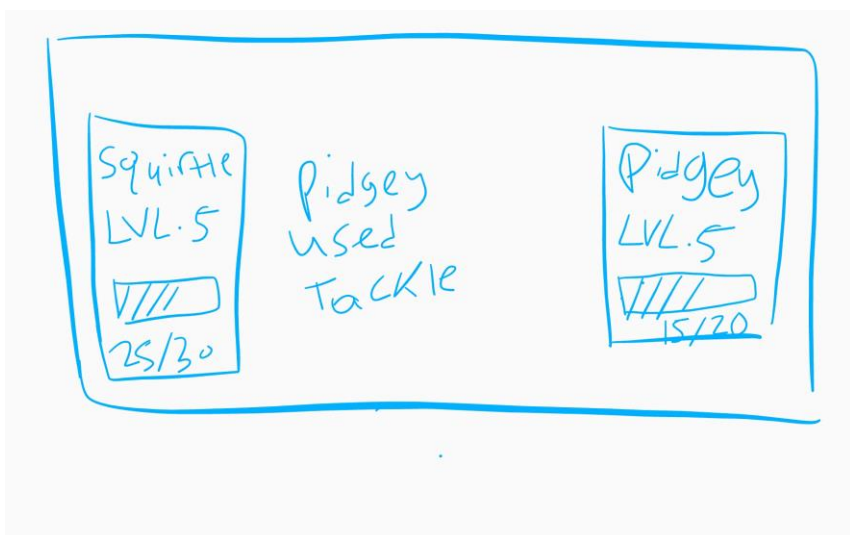
Now we see a battle screen that shows information about your Pokémon on the left side and the other Pokémon on the right side. There is a health bar indicating HP (hitpoints) and a Level. The four options are fight, bag, Pokémon, run. This battle scenario will not cover the run command as it simply goes back to the previous navigation (in this case, inside grass).



Upon clicking on fight, there are a maximum of 4 attacks. Return will change the four buttons to fight, bag, Pokémon, and run once again.



Upon clicking on Tackle, The buttons disappear and text appears. The health bar is reflected based upon damage calculations.

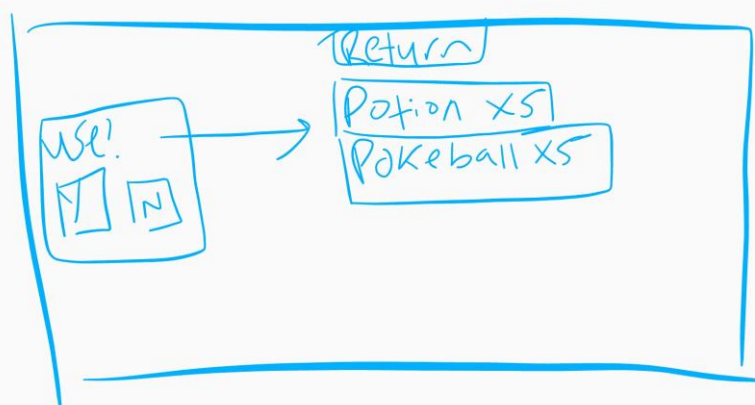


It is now the other Pokémon's turn.

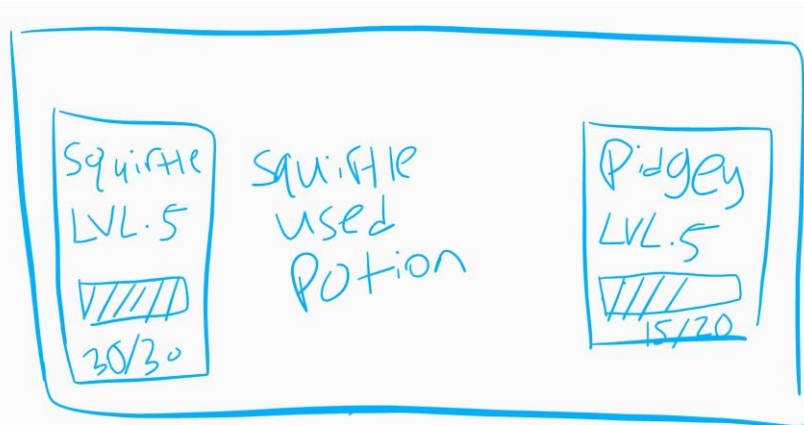


Now we see the main battle menu once again. The next method to show is bag in order to use an item to heal our Squirtle.

## UseItem



Upon clicking on bag, items are shown. If there are too many items, there will be a “next page button”. Clicking on an item will bring a menu towards the left/right side. This menu will have two buttons. Yes, will use the item and no will make this submenu disappear.

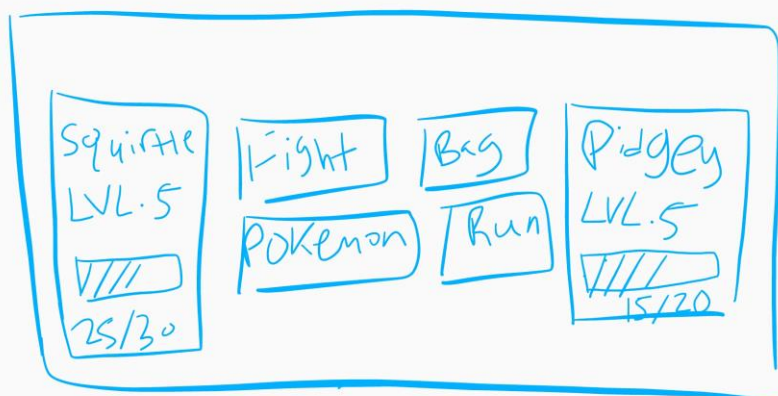


If one were to use a potion, player's Pokémon's HP will increase.

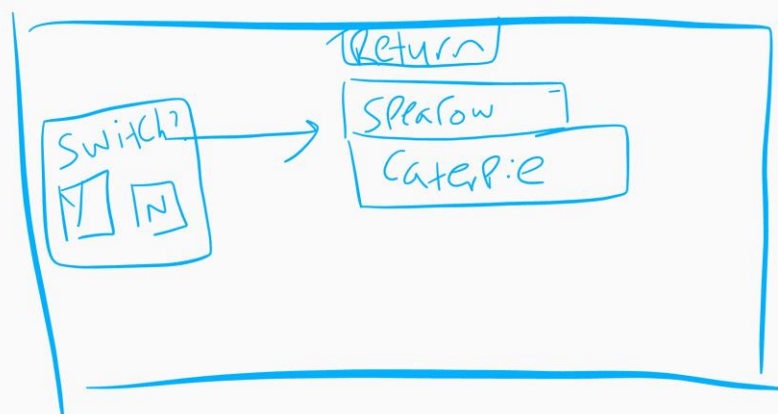


Again it is now the other Pokémon's turn.

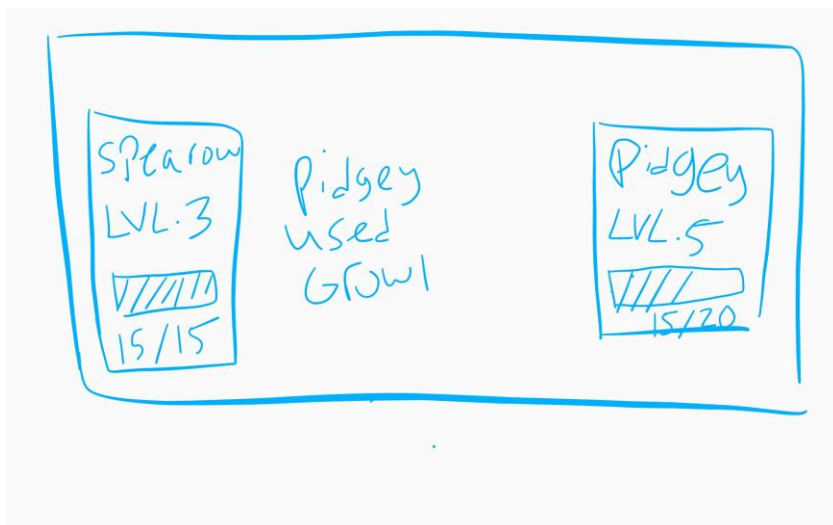




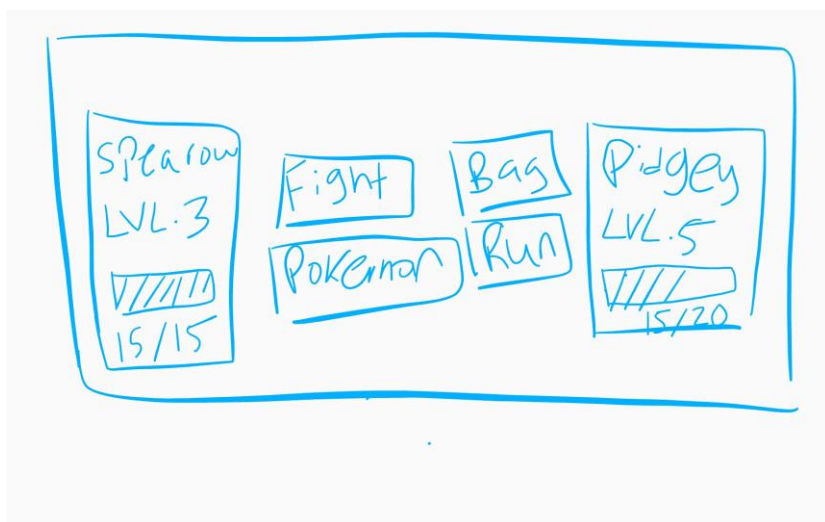
The main battle menu is now shown again. The next method to discuss is switching Pokémon with another one in the user's party (maximum of 6).



Clicking on the Pokémon button brings a similar menu in comparison with the bag button. The only difference is instead of “use”, we have “switch” and Pokémon are displayed instead of items. **(Need to add summary option as well)**

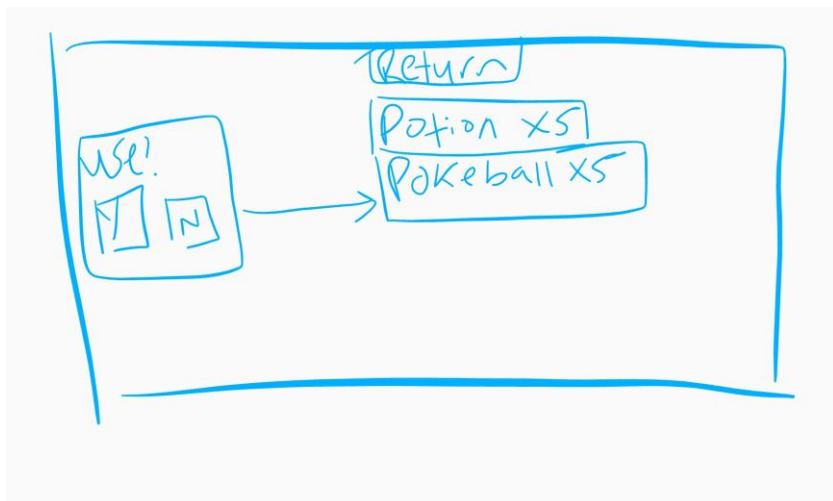


When a Pokémon is switched, the different name, level, and HP is displayed. It is now the other Pokémon's turn however.

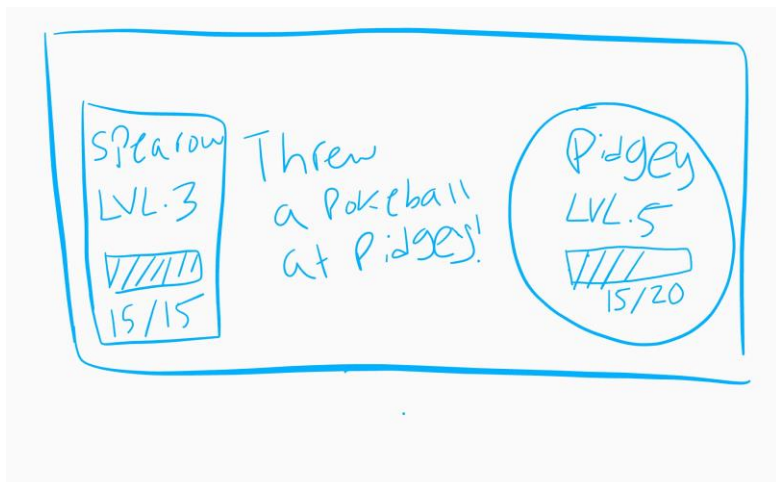


Again, the main battle menu is shown. Next we will discuss how to catch a Pokémon.

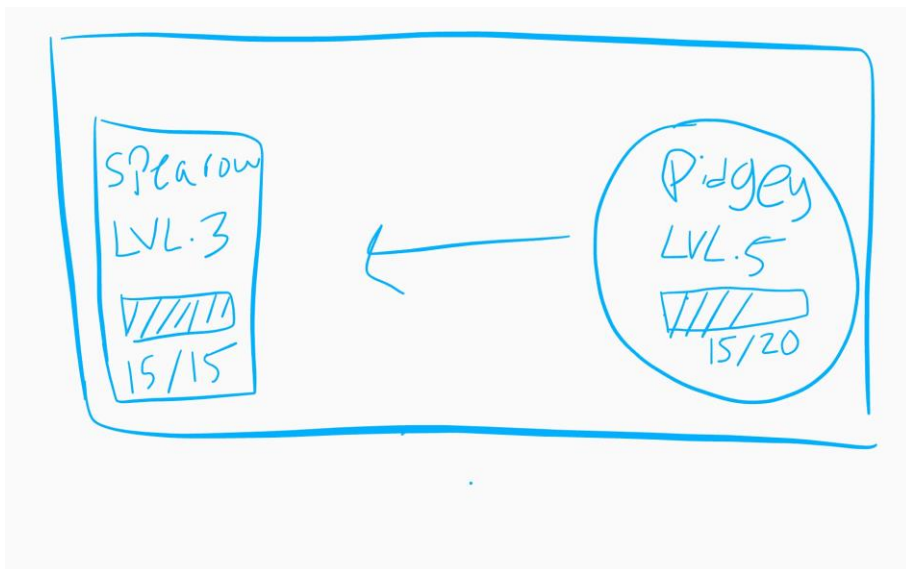
## Catch



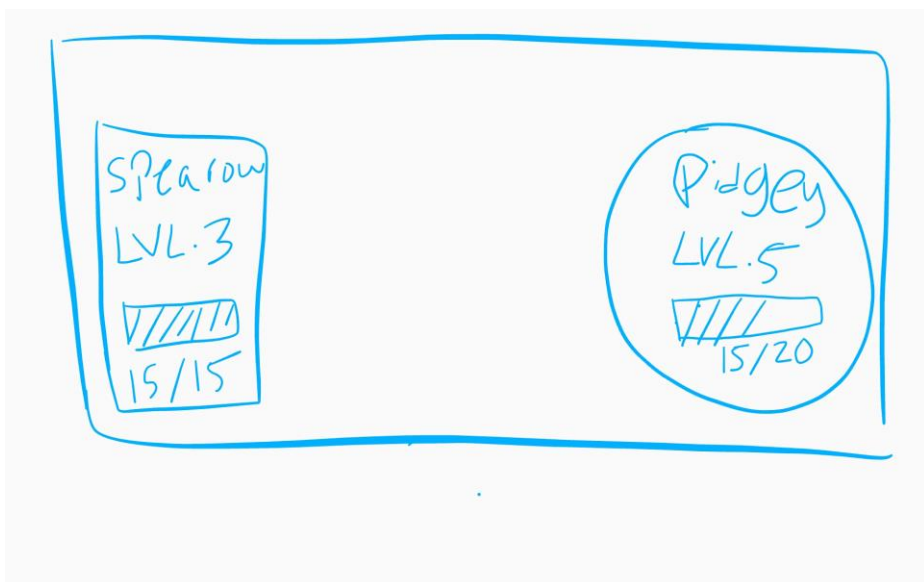
In order to catch a Pokémon, one would have to go into the bag and use a Pokéball.



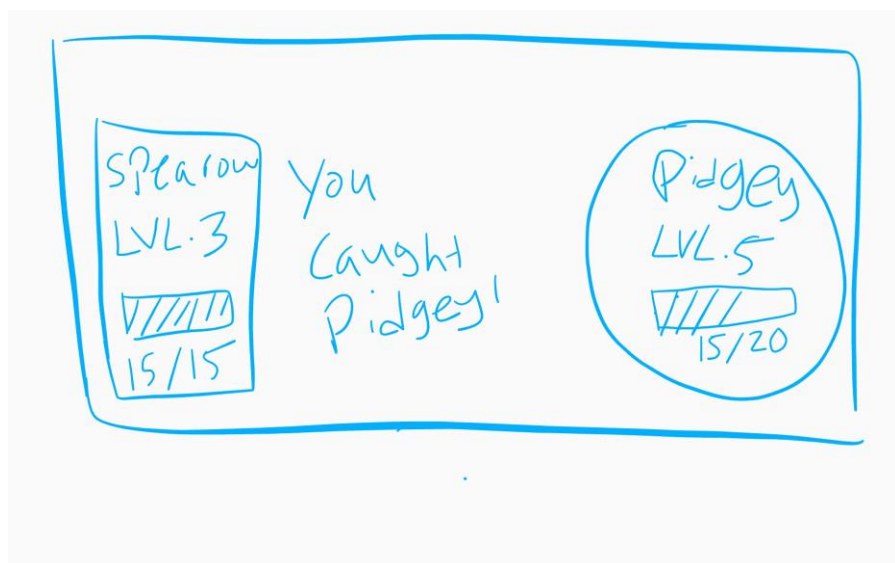
The panel for the other Pokémon will now have a circle surrounding it.



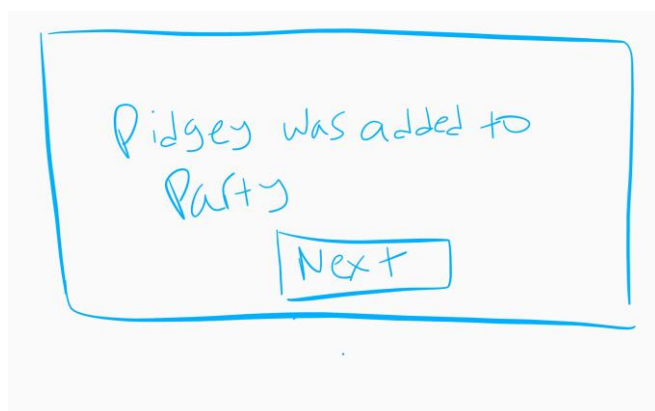
The entire panel for the other Pokémon will move towards the left.



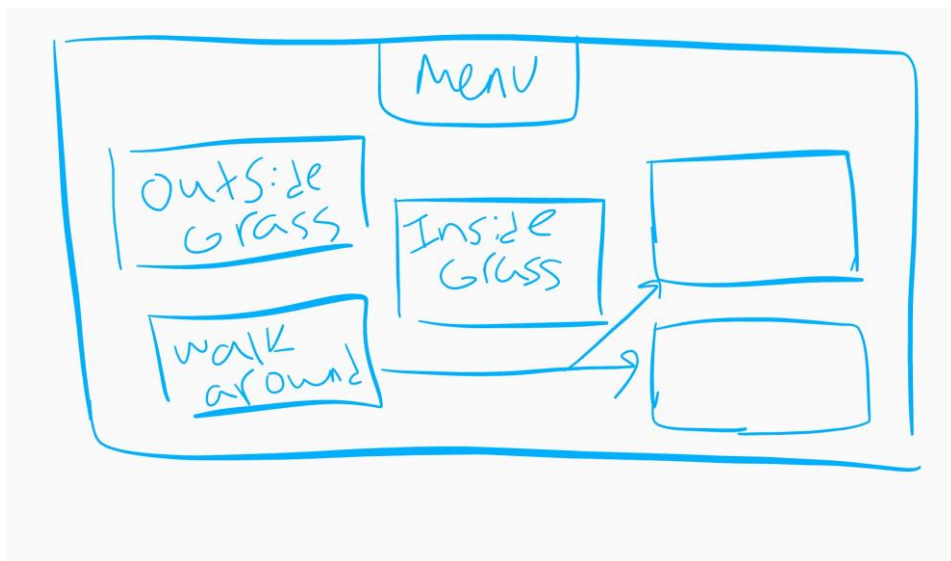
Then the panel moves towards the right.



Finally, the Pokémon is caught.



Now, information about the Pokémon is displayed with a button to allow user to proceed. In a trainer battle, another screen will be shown afterwards with a quick dialogue following it. Upon defeating a wild Pokémon, another screen will be shown including amount of experience points earned, money earned, or increase of any levels.



After a battle summary of any kind, return to the main navigation.

## User Effort Estimation

**Note:** All button clicks will have a keystroke

### Scenario 1

(Above Example)

#### 1. Navigation (Total 5 Mouse clicks & 5 Keystrokes)

- a. Click on Route 1
- b. Click on Outside Grass
- c. Click on Inside Grass
- d. Click walk around (however many times necessary)
- e. Click Next

#### 2. Data Entry (Total 12 mouse clicks & 12 Keystrokes)

- a. Click on Fight
- b. Click on Tackle
- c. Click on Bag
- d. Click on Potion
- e. Click Yes
- f. Click on Pokémon
- g. Click on Spearow
- h. Click Yes
- i. Click on Bag
- j. Click on Pokéball
- k. Click Yes
- l. Click Next

## Scenario 2

**(From Pallet Town to Pokémon Center in Viridian City in order to deposit 1 Pokémon and withdraw 1 Pokémon)**

### **1. Navigation (Total 4 mouse clicks & 4 Keystrokes)**

- a. Click on Route 1
- b. Click on Viridian City
- c. Click on Pokémon Center
- d. Click on PC

### **2. Data Entry (Total 8 mouse clicks & 8 Keystrokes)**

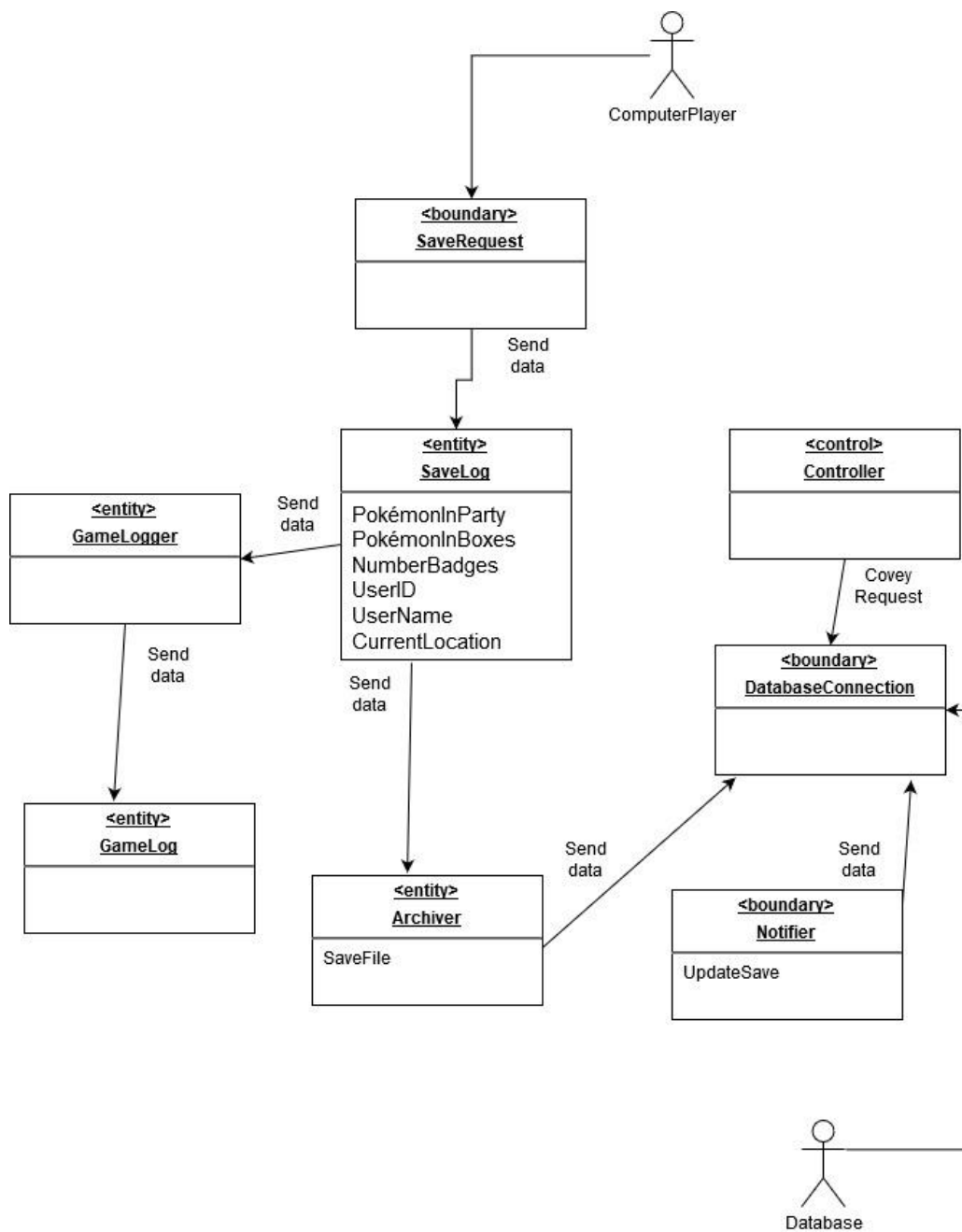
- a. Click on Pokémon
- b. Click on Deposit
- c. Click on (Pokémon name)
- d. Click on Yes
- e. Click on Return
- f. Click on Withdraw
- g. Click on (Pokémon name)
- h. Click on Yes



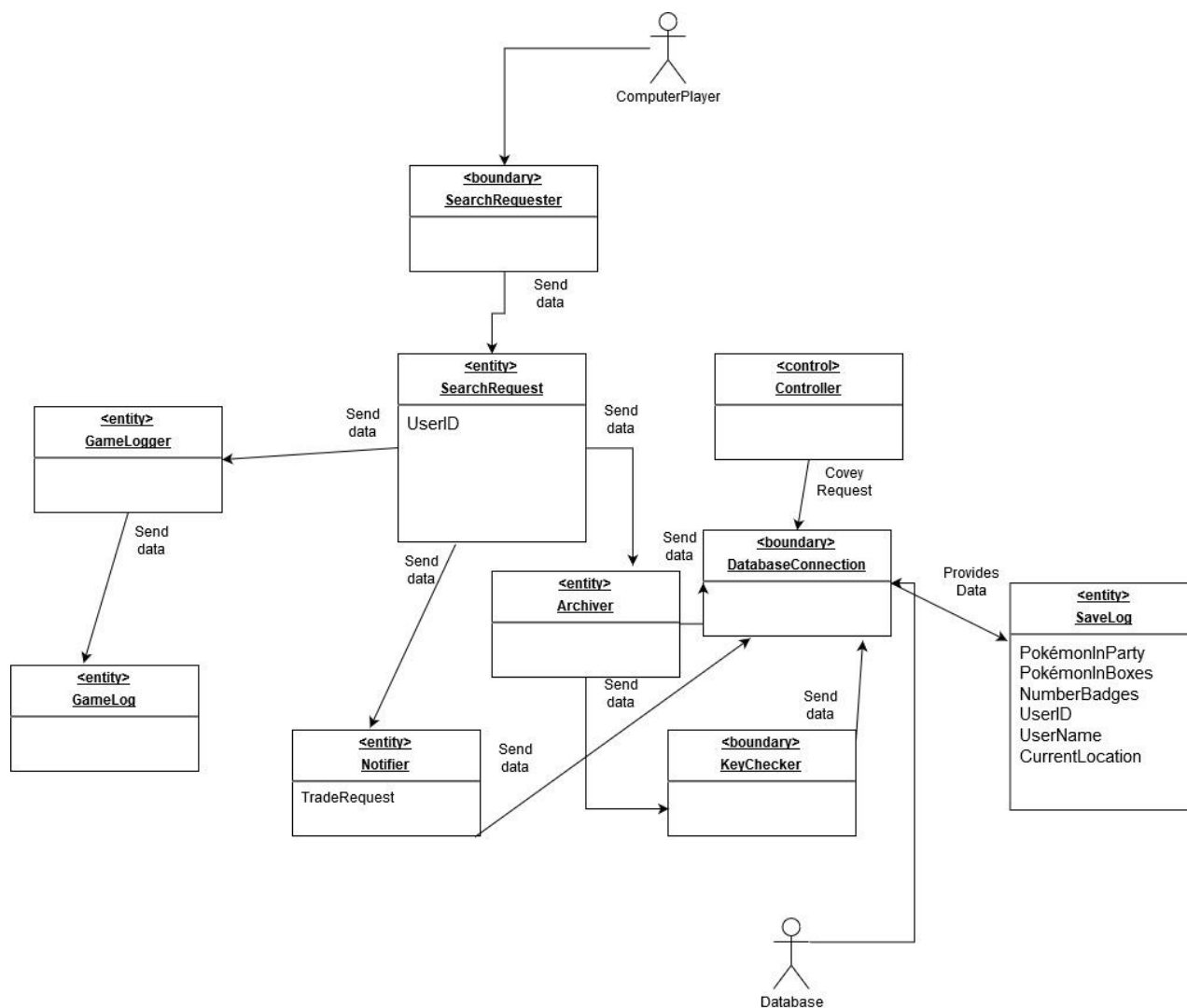
## Domain Analysis

### Domain Model

#### SaveGame (UC-3)



## StartTradeRequest (UC-5)



## Concept Definition

Responsibility Description	Type	Concept Name
Controls coordination of concepts within use cases or the entire system as a whole.	D	Controller
Contains information about player's current game state	K	SaveLog
checks if input key is valid	D	KeyChecker
Stores all information as text as game progresses	D	GameLogger
Contains all information as game progresses	K	GameLog
Outputs GameLog as speech	D	TextToSpeech
Checks for max inventory space	D	MaxChecker
Search database based off of ID of other player's save file	D	SearchRequester
Contains information about database ID search	K	SearchRequest
Outputs external URL of feedback page	D	PageMaker
Archives saves in database	D	Archiver
Form containing feedback information	K	FeedbackPage
Notify user about all accepted/failed requests	D	Notifier
Outputs sound	D	OutputSound
Connects to Database	D	DatabaseConnection
Save to save file	D	SaveRequest

## Association Definitions

Concept Pair	Association Description	Association Name
SaveLog - Archiver	SaveLog sends data to Archiver	Sends Data
Archiver - SaveLog	Archiver retrieves data from SaveLog	Retrieves Data
SearchRequester - Archiver	SearchRequester sends data to Archiver	Sends Data
GameLogger - GameLog	GameLogger sends data to GameLog	Sends data
TextToSpeech - GameLog	TextToSpeech Retrieves Data from GameLog	Retrieves Data From
OutputSound – TextToSpeech	OutputSound retrieves data from TextToSpeech	Retrieves Data from
PageMaker – FeedbackPage	PageMaker prepares FeedbackPage	Prepares
Notifier - MaxChecker	Notifier checks MaxChecker	Checks
Notifier - KeyChecker	Notifier checks KeyChecker	Checks
Controller – PageMaker	Controller conveys request to PageMaker	Conveys Request
Controller – SearchRequester	Controller conveys request to SearchRequester	Conveys Request
Controller – DatabaseConnection	Controller conveys request to DatabaseConnection	Conveys Request
Notifier-DatabaseConnection	Notifier sends data to DatabaseConnection	Sends data
SaveRequest - Archiver	SaveRequest sends data to Archiver	Sends data
SaveRequest - GameLogger	SaveRequest sends data to GameLogger	Sends data
DatabaseConnection - SaveLog	DatabaseConnection provides data for SaveLog	Provides data

## Attribute Definitions

Save Log	Pokémon In Party	Contains core information such as names, levels, stats, moves
	Pokémon In Boxes	Contains core information such as names, levels, stats, moves, box number
	Number of Gym Badges Obtained	Essentially game completion progress, there are 8 gym badges
	User ID	Random ID created from initializing/starting a game
	User Name	Name that user chooses at start of game
	Current Location	Location that user saves from
MaxChecker	Max Exceeded	Max Pokémon in party is 6, Max Pokémon in PC Box is 30, Max Items in bag is 50, Max items in PC Box is 100
KeyChecker	User ID	Random ID created from initializing/starting a game
Notifier	Trade Request	Random ID created from initializing/starting a game
PageMaker	URL	The specific URL for the page
Archiver	SaveFile	The save text file
SearchRequest	UserID	Random ID created from initializing/starting a game

## Traceability Matrix

							Domain Concepts											
Use Case	P W	Controller	SaveLog	KeyChecker	GameLogger	GameLog	TextToSpeech	MaxChecker	SearchRequester	SearchRequest	PageMaker	Archiver	FeedbackPage	Notifier	OutputSound	DatabaseConnection	SaveRequest	
UC1	3	X			X	X												
UC2	7	X			X	X												
UC3	2	X	X		X	X						X		X		X	X	
UC4	2	X	X		X	X								X				
UC5	4	X	X	X	X	X			X	X		X		X		X		
UC6	5	X			X	X								X				
UC7	1	X			X	X					X		X					
UC8	5	X			X	X												
UC9	14	X			X	X		X										
UC10	3	X			X	X												
UC11	4	X			X	X												

UC1: StartGame

UC2: NavigateGame

UC3: SaveGame

UC4: LoadGame

UC5: StartTradeRequest

UC6: UseItem

UC7: Survey

UC8: Attack

UC9: Catch

UC10: Heal

UC11: TrainerBattle

## **System Operation Contracts**

## Mathematical Model

Experience points measures the amount of experience an individual Pokémon possesses, which was gained in battle in defeating an opponent Pokémon without fainting. After a certain amount of experience points have been gained, a Pokémon will grow a level, all the way up to level 100, where a Pokémon will no longer gain experience.

The amount of experience points a Pokémon has is tied directly to its level. The four functions, the ones for the Fast, Medium Fast, Medium Slow, and Slow groups, are cubic. The two that were added in Generation III (Erratic and Fluctuating), however, are made by multiplying the cube of the level by a linear function of it (a negative slope one in the case of Erratic, going from 2 to 0.6; and a positive slope one in the case of Fluctuating, going from 0.48 to 1.64), making those functions effectively quartic.

Level-up rates can be calculated using an equation, a lookup table is used in the games to prevent game slowdown and a glitch associated with the Medium Slow formula.

	Formula
Erratic	$EXP = \begin{cases} \frac{n^3(100-n)}{50} & n \leq 50 \\ \frac{n^3(150-n)}{50} & 50 \leq n \leq 68 \\ \frac{n^3 \lfloor \frac{100}{3} - \frac{10n}{3} \rfloor}{500} & 68 \leq n \leq 98 \\ \frac{n^3(160-n)}{100} & 98 \leq n \leq 100 \end{cases}$
Fast	$EXP = \frac{4n^3}{5}$
Medium Fast	$EXP = n^3$
Medium Slow	$EXP = \frac{6}{5}n^3 - 15n^2 + 100n - 140$
Slow	$EXP = \frac{5n^3}{4}$
Fluctuating	$EXP = \begin{cases} n^3 \left( \frac{\lfloor \frac{n+1}{3} \rfloor + 24}{50} \right) & n \leq 15 \\ n^3 \left( \frac{n+14}{50} \right) & 15 \leq n \leq 36 \\ n^3 \left( \frac{\lfloor \frac{n}{2} \rfloor + 32}{50} \right) & 36 \leq n \leq 100 \end{cases}$



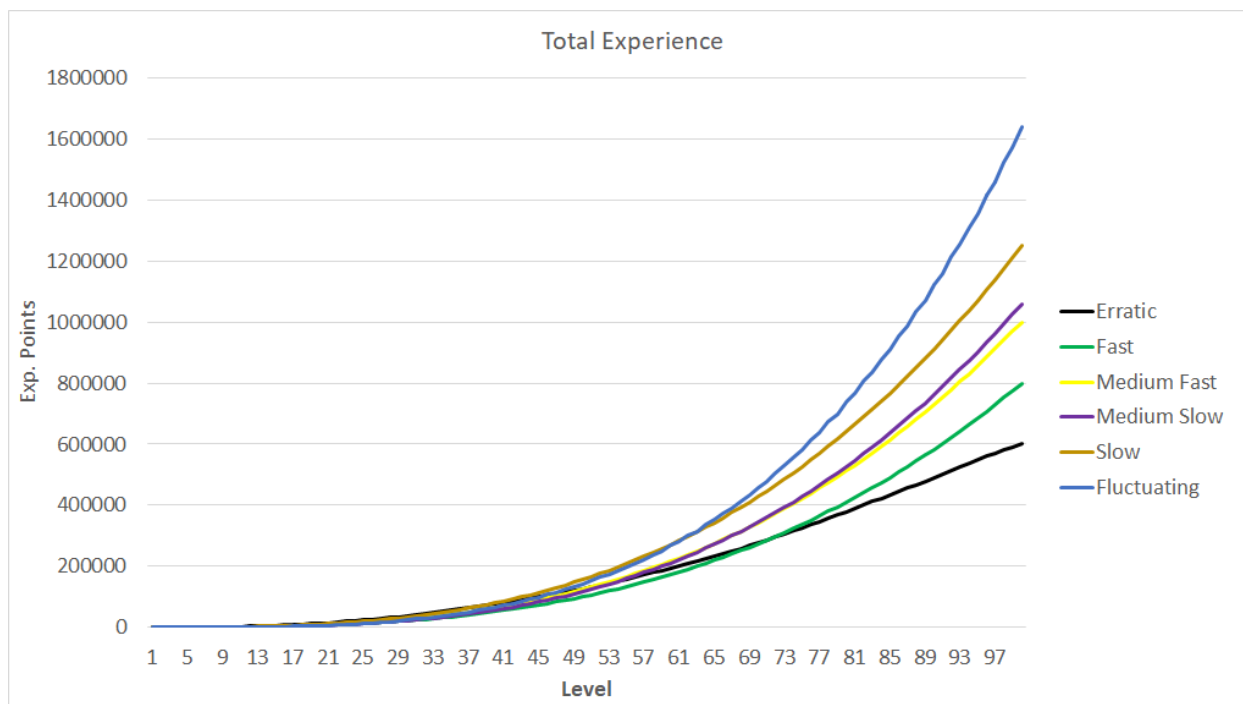
Below is a table; on the left side of the level is the minimum number of experience points required for a Pokémon to be at that level, and the amount of experience points a Pokémon of that level will have when caught from the wild; on the right is the number of experience points required to advance from the respective level to the next level.

Experience table													
Total experience							To next level						
Level	Erratic	Fast	Medium Fast	Medium Slow	Slow	Fluctuating	Level	Erratic	Fast	Medium Fast	Medium Slow	Slow	Fluctuating
1	0	0	0	0	0	0	1	15	6	8	9	10	4
2	15	6	8	9	10	4	2	37	15	19	48	23	9
3	52	21	27	57	33	13	3	70	30	37	39	47	19
4	122	51	64	96	80	32	4	115	49	61	39	76	33
5	237	100	125	135	156	65	5	169	72	91	44	114	47
6	406	172	216	179	270	112	6	231	102	127	57	158	66
7	637	274	343	236	428	178	7	305	135	169	78	212	98
8	942	409	512	314	640	276	8	384	174	217	105	271	117
9	1,326	583	729	419	911	393	9	474	217	271	141	339	147
10	1,800	800	1,000	560	1,250	540	10	569	264	331	182	413	205
11	2,369	1,064	1,331	742	1,663	745	11	672	318	397	231	497	222
12	3,041	1,382	1,728	973	2,160	967	12	781	375	469	288	586	263
13	3,822	1,757	2,197	1,261	2,746	1,230	13	897	438	547	351	684	361
14	4,719	2,195	2,744	1,612	3,430	1,591	14	1,018	505	631	423	788	366
15	5,737	2,700	3,375	2,035	4,218	1,957	15	1,144	576	721	500	902	500
16	6,881	3,276	4,096	2,535	5,120	2,457	16	1,274	654	817	585	1,021	589
17	8,155	3,930	4,913	3,120	6,141	3,046	17	1,409	735	919	678	1,149	686
18	9,564	4,665	5,832	3,798	7,290	3,732	18	1,547	822	1,027	777	1,283	794
19	11,111	5,487	6,859	4,575	8,573	4,526	19	1,689	913	1,141	885	1,427	914
20	12,800	6,400	8,000	5,460	10,000	5,440	20	1,832	1,008	1,261	998	1,576	1,042
21	14,632	7,408	9,261	6,458	11,576	6,482	21	1,978	1,110	1,387	1,119	1,734	1,184
22	16,610	8,518	10,648	7,577	13,310	7,666	22	2,127	1,215	1,519	1,248	1,898	1,337
23	18,737	9,733	12,167	8,825	15,208	9,003	23	2,275	1,326	1,657	1,383	2,072	1,503
24	21,012	11,059	13,824	10,208	17,280	10,506	24	2,425	1,441	1,801	1,527	2,251	1,681
25	23,437	12,500	15,625	11,735	19,531	12,187	25	2,575	1,560	1,951	1,676	2,439	1,873
26	26,012	14,060	17,576	13,411	21,970	14,060	26	2,725	1,686	2,107	1,833	2,633	2,080
27	28,737	15,746	19,683	15,244	24,603	16,140	27	2,873	1,815	2,269	1,998	2,837	2,299
28	31,610	17,561	21,952	17,242	27,440	18,439	28	3,022	1,950	2,437	2,169	3,046	2,535
29	34,632	19,511	24,389	19,411	30,486	20,974	29	3,168	2,089	2,611	2,349	3,264	2,786
30	37,800	21,600	27,000	21,760	33,750	23,760	30	3,311	2,232	2,791	2,534	3,488	3,051
31	41,111	23,832	29,791	24,294	37,238	26,811	31	3,453	2,382	2,977	2,727	3,722	3,335
32	44,564	26,214	32,768	27,021	40,960	30,146	32	3,591	2,535	3,169	2,928	3,961	3,634
33	48,155	28,749	35,937	29,949	44,921	33,780	33	3,726	2,694	3,367	3,135	4,209	3,951

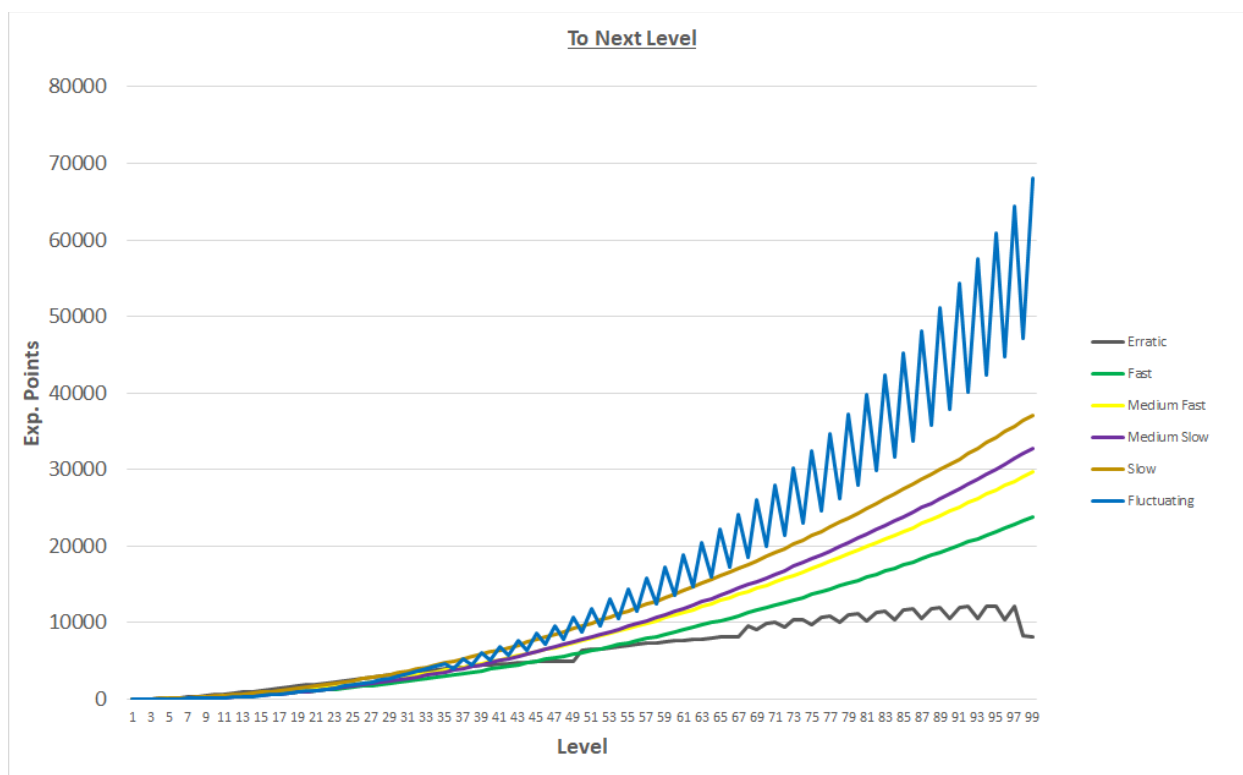
34	51,881	31,443	39,304	33,084	49,130	37,731	34	3,856	2,857	3,571	3,351	4,463	4,286
35	55,737	34,300	42,875	36,435	53,593	42,017	35	3,982	3,024	3,781	3,572	4,727	4,639
36	59,719	37,324	46,656	40,007	58,320	46,656	36	4,103	3,198	3,997	3,801	4,996	3,997
37	63,822	40,522	50,653	43,808	63,316	50,653	37	4,219	3,375	4,219	4,038	5,274	5,316
38	68,041	43,897	54,872	47,846	68,590	55,969	38	4,328	3,558	4,447	4,281	5,558	4,536
39	72,369	47,455	59,319	52,127	74,148	60,505	39	4,431	3,745	4,681	4,533	5,852	6,055
40	76,800	51,200	64,000	56,660	80,000	66,560	40	4,526	3,936	4,921	4,790	6,151	5,117
41	81,326	55,136	68,921	61,450	86,151	71,677	41	4,616	4,134	5,167	5,055	6,459	6,856
42	85,942	59,270	74,088	66,505	92,610	78,533	42	4,695	4,335	5,419	5,328	6,773	5,744
43	90,637	63,605	79,507	71,833	99,383	84,277	43	4,769	4,542	5,677	5,607	7,097	7,721
44	95,406	68,147	85,184	77,440	106,480	91,998	44	4,831	4,753	5,941	5,895	7,426	6,417
45	100,237	72,900	91,125	83,335	113,906	98,415	45	4,885	4,968	6,211	6,188	7,764	8,654
46	105,122	77,868	97,336	89,523	121,670	107,069	46	4,930	5,190	6,487	6,489	8,108	7,136
47	110,052	83,058	103,823	96,012	129,778	114,205	47	4,963	5,415	6,769	6,798	8,462	9,658
48	115,015	88,473	110,592	102,810	138,240	123,863	48	4,986	5,646	7,057	7,113	8,821	7,903
49	120,001	94,119	117,649	109,923	147,061	131,766	49	4,999	5,881	7,351	7,437	9,189	10,734
50	125,000	100,000	125,000	117,360	156,250	142,500	50	6,324	6,120	7,651	7,766	9,563	8,722
51	131,324	106,120	132,651	125,126	165,813	151,222	51	6,471	6,366	7,957	8,103	9,947	11,883
52	137,795	112,486	140,608	133,229	175,760	163,105	52	6,615	6,615	8,269	8,448	10,336	9,592
53	144,410	119,101	148,877	141,677	186,096	172,697	53	6,755	6,870	8,587	8,799	10,734	13,110
54	151,165	125,971	157,464	150,476	196,830	185,807	54	6,891	7,129	8,911	9,159	11,138	10,515
55	158,056	133,100	166,375	159,635	207,968	196,322	55	7,023	7,392	9,241	9,524	11,552	14,417
56	165,079	140,492	175,616	169,159	219,520	210,739	56	7,150	7,662	9,577	9,897	11,971	11,492
57	172,229	148,154	185,193	179,056	231,491	222,231	57	7,274	7,935	9,919	10,278	12,399	15,805
58	179,503	156,089	195,112	189,334	243,890	238,036	58	7,391	8,214	10,267	10,665	12,833	12,526
59	186,894	164,303	205,379	199,999	256,723	250,562	59	7,506	8,497	10,621	11,061	13,277	17,278
60	194,400	172,800	216,000	211,060	270,000	267,840	60	7,613	8,784	10,981	11,462	13,726	13,616
61	202,013	181,584	226,981	222,522	283,726	281,456	61	7,715	9,078	11,347	11,871	14,184	18,837
62	209,728	190,662	238,328	234,393	297,910	300,293	62	7,812	9,375	11,719	12,288	14,648	14,766
63	217,540	200,037	250,047	246,681	312,558	315,059	63	7,903	9,678	12,097	12,711	15,122	20,485
64	225,443	209,715	262,144	259,392	327,680	335,544	64	7,988	9,985	12,481	13,143	15,601	15,976
65	233,431	219,700	274,625	272,535	343,281	351,520	65	8,065	10,296	12,871	13,580	16,089	22,224
66	241,496	229,996	287,496	286,115	359,370	373,744	66	8,137	10,614	13,267	14,025	16,583	17,247
67	249,633	240,610	300,763	300,140	375,953	390,991	67	8,201	10,935	13,669	14,478	17,087	24,059
68	257,834	251,545	314,432	314,618	393,040	415,050	68	9,572	11,262	14,077	14,937	17,596	18,581
69	267,406	262,807	328,509	329,555	410,636	433,631	69	9,052	11,593	14,491	15,405	18,114	25,989
70	276,458	274,400	343,000	344,960	428,750	459,620	70	9,870	11,928	14,911	15,878	18,638	19,980
71	286,328	286,328	357,911	360,838	447,388	479,600	71	10,030	12,270	15,337	16,359	19,172	28,017
72	296,358	298,598	373,248	377,197	466,560	507,617	72	9,409	12,615	15,769	16,848	19,711	21,446
73	305,767	311,213	389,017	394,045	486,271	529,063	73	10,307	12,966	16,207	17,343	20,259	30,146
74	316,074	324,179	405,224	411,388	506,530	559,209	74	10,457	13,321	16,651	17,847	20,813	22,978
75	326,531	337,500	421,875	429,235	527,343	582,187	75	9,724	13,680	17,101	18,356	21,377	32,379

76	336,255	351,180	438,976	447,591	548,720	614,566	76	10,710	14,046	17,557	18,873	21,946	24,580
77	346,965	365,226	456,533	466,464	570,666	639,146	77	10,847	14,415	18,019	19,398	22,524	34,717
78	357,812	379,641	474,552	485,862	593,190	673,863	78	9,995	14,790	18,487	19,929	23,108	26,252
79	367,807	394,431	493,039	505,791	616,298	700,115	79	11,073	15,169	18,961	20,469	23,702	37,165
80	378,880	409,600	512,000	526,260	640,000	737,280	80	11,197	15,552	19,441	21,014	24,301	27,995
81	390,077	425,152	531,441	547,274	664,301	765,275	81	10,216	15,942	19,927	21,567	24,909	39,722
82	400,293	441,094	551,368	568,841	689,210	804,997	82	11,393	16,335	20,419	22,128	25,523	29,812
83	411,686	457,429	571,787	590,969	714,733	834,809	83	11,504	16,734	20,917	22,695	26,147	42,392
84	423,190	474,163	592,704	613,664	740,880	877,201	84	10,382	17,137	21,421	23,271	26,776	31,704
85	433,572	491,300	614,125	636,935	767,656	908,905	85	11,667	17,544	21,931	23,852	27,414	45,179
86	445,239	508,844	636,056	660,787	795,070	954,084	86	11,762	17,958	22,447	24,441	28,058	33,670
87	457,001	526,802	658,503	685,228	823,128	987,754	87	10,488	18,375	22,969	25,038	28,712	48,083
88	467,489	545,177	681,472	710,266	851,840	1,035,837	88	11,889	18,798	23,497	25,641	29,371	35,715
89	479,378	563,975	704,969	735,907	881,211	1,071,552	89	11,968	19,225	24,031	26,253	30,039	51,108
90	491,346	583,200	729,000	762,160	911,250	1,122,660	90	10,532	19,656	24,571	26,870	30,713	37,839
91	501,878	602,856	753,571	789,030	941,963	1,160,499	91	12,056	20,094	25,117	27,495	31,397	54,254
92	513,934	622,950	778,688	816,525	973,360	1,214,753	92	12,115	20,535	25,669	28,128	32,086	40,043
93	526,049	643,485	804,357	844,653	1,005,446	1,254,796	93	10,508	20,982	26,227	28,767	32,784	57,526
94	536,557	664,467	830,584	873,420	1,038,230	1,312,322	94	12,163	21,433	26,791	29,415	33,488	42,330
95	548,720	685,900	857,375	902,835	1,071,718	1,354,652	95	12,202	21,888	27,361	30,068	34,202	60,925
96	560,922	707,788	884,736	932,903	1,105,920	1,415,577	96	10,411	22,350	27,937	30,729	34,921	44,699
97	571,333	730,138	912,673	963,632	1,140,841	1,460,276	97	12,206	22,815	28,519	31,398	35,649	64,455
98	583,539	752,953	941,192	995,030	1,176,490	1,524,731	98	8,343	23,286	29,107	32,073	36,383	47,153
99	591,882	776,239	970,299	1,027,103	1,212,873	1,571,884	99	8,118	23,761	29,701	32,757	37,127	68,116
100	600,000	800,000	1,000,000	1,059,860	1,250,000	1,640,000	100	-	-	-	-	-	-

Below is a graph of the experience required for a Pokémon to be a certain level, color-coded by experience types. Erratic is **black**, Fast is **green**, Medium Fast is **yellow**, Medium Slow is **purple**, Slow is **brown**, and Fluctuating is **blue**.



Below graph showing experience needed to gain a single level, for each level up to 100. The nature of the Erratic and Fluctuating curves can be seen more clearly here.



## Project Size Estimation

---

# Plan of Work

---

## September 23-29:

- Finish "Interactive Diagrams" for Report #2
- Set up GUI for each screen (all locations, main game screen, options, etc.) (each screen except for specific locations listed in User Interface Requirements)

## September 30- October 6:

- Finish "Class diagram and system architecture" for Report #2
- Set up Items superclass with subclasses (Aim for 20 items)
- Set up Pokémon superclass with subclasses (Aim for 20 Pokémon)
- Set up NPC trainers superclass with subclasses (Aim for 20 Trainers)
- Set up Attacks superclass with subclasses (Aim for 50 attacks)

## October 7-13:

- Finish Report #2
- Link locations class to Pokémon objects for wild encounters using Java random
- Link NPC trainers class to Pokémon objects (give trainers Pokémon)
- Start battle system (fight, useItem, run, and switch Pokémon)

## October 14-20:

- Finish battle system

## October 21-27:

- Piece the game together with locations, story, and NPC Trainers

## October 28-November 3:

- Present First Demo of Battle System by itself
- Piece the game together with locations, story, and NPC Trainers
- Incorporate soundtrack
  - With soundtrack off option in main settings
- Incorporate text to sound

## November 4-10:

- Incorporate Trade mechanic

**November 11-17:**

- Finish part one of Report #3
- Finish up incorporating system options such as text speed and anything else not finished

**November 18-24:**

- Finish Report 3 (add both Report 1 and Report 2)
- Test product

**November 25-December 1:**

- Present Final Product

## References

---

Bulbapedia.bulbagarden.net. (2019). *Statistic - Bulbapedia, the community-driven Pokémon encyclopedia*. [online] Available at: <https://bulbapedia.bulbagarden.net/wiki/Statistic> [Accessed 20 Sep. 2019].

Bulbapedia.bulbagarden.net. (2019). *Catch rate - Bulbapedia, the community-driven Pokémon encyclopedia*. [online] Available at: [https://bulbapedia.bulbagarden.net/wiki/Catch\\_rate](https://bulbapedia.bulbagarden.net/wiki/Catch_rate) [Accessed 20 Sep. 2019].

Downloads.khinsider.com. (2016). *Pokemon Red, Green, Blue & Yellow MP3 - Download Pokemon Red,*

Gamefaqs.gamespot.com. (2007). *Pokemon Red Version - Game Script - Game Boy - By mtkennerly - GameFAQs*. [online] Available at: <https://gamefaqs.gamespot.com/gameboy/367023-pokemon-red-version/faqs/48982> [Accessed 20 Sep. 2019].

Gamefaqs.gamespot.com. (2019). *Pokemon Blue Version Box Shot for Game Boy - GameFAQs*. [online] Available at: <https://gamefaqs.gamespot.com/gameboy/907714-pokemon-blue-version/images/168475> [Accessed 20 Sep. 2019].

*Green, Blue & Yellow Soundtracks for FREE!*. [online] Available at: <https://downloads.khinsider.com/game-soundtracks/album/pokemon-red-green-blue-yellow> [Accessed 20 Sep. 2019].

M.bulbapedia.bulbagarden.net. (2019). *Experience - Bulbapedia, the community-driven Pokémon encyclopedia*. [online] Available at: <https://m.bulbapedia.bulbagarden.net/wiki/Experience> [Accessed 20 Sep. 2019].

Marsic, I. (2012). *Software Engineering*. New Brunswick.

MobyGames. (2018). *Pokémon Blue Version (1998) Game Boy box cover art - MobyGames*. [online] Available at: <https://www.mobygames.com/game/pokmon-blue-version/cover-art/gameCoverId,453392/> [Accessed 20 Sep. 2019].

App.creately.com. (2019). *Creately*. [online] Available at: <https://app.creately.com/diagram/iREP4saBgEy/edit> [Accessed 20 Sep. 2019].