Project in Information Management

Presented to the Faculty of Computer Department

for the course of

Bachelor of Science in Information Technology (BSIT)

Ken Adrien E. Arceno

Nimfa Taupo

College Faculty

March 2025

Name of Database Project: PokéDexDB

Target Users: Players, Game Developers, and Researchers focusing on Pokémon FireRed game mechanics

Description: PokéDexDB is a relational database designed to store and manage core game data from Pokémon FireRed, a Generation 1-based Pokémon game. It allows users to query detailed information on Pokémon's stats, learnable moves, trainers, locations etc.. The primary goal is to create a structured and efficient database system that reflects the original game mechanics while supporting data analysis and visualization.

Design Concepts

1. Database Design using Class Diagram

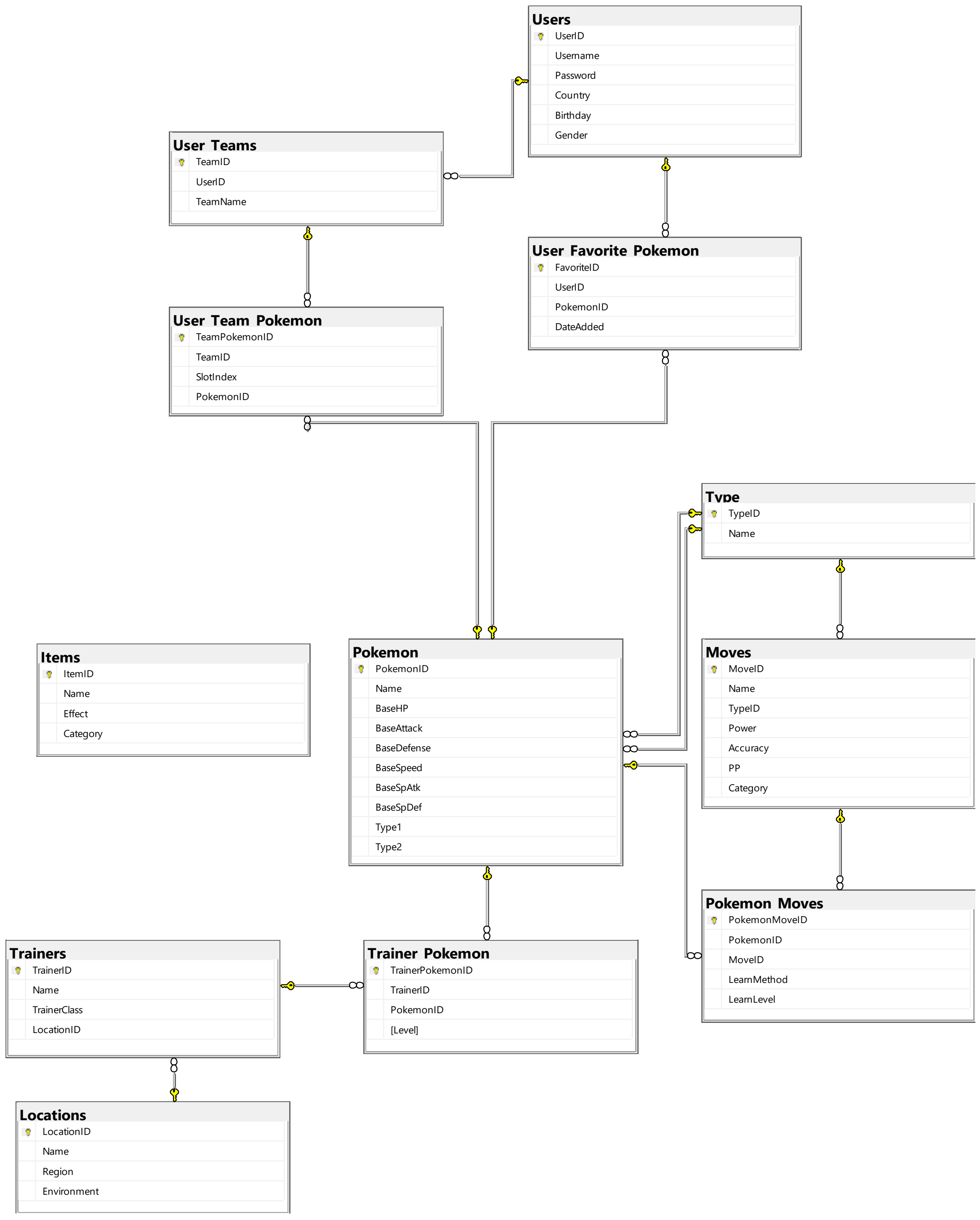


Figure 1: Design Diagram with Structure in SSMS

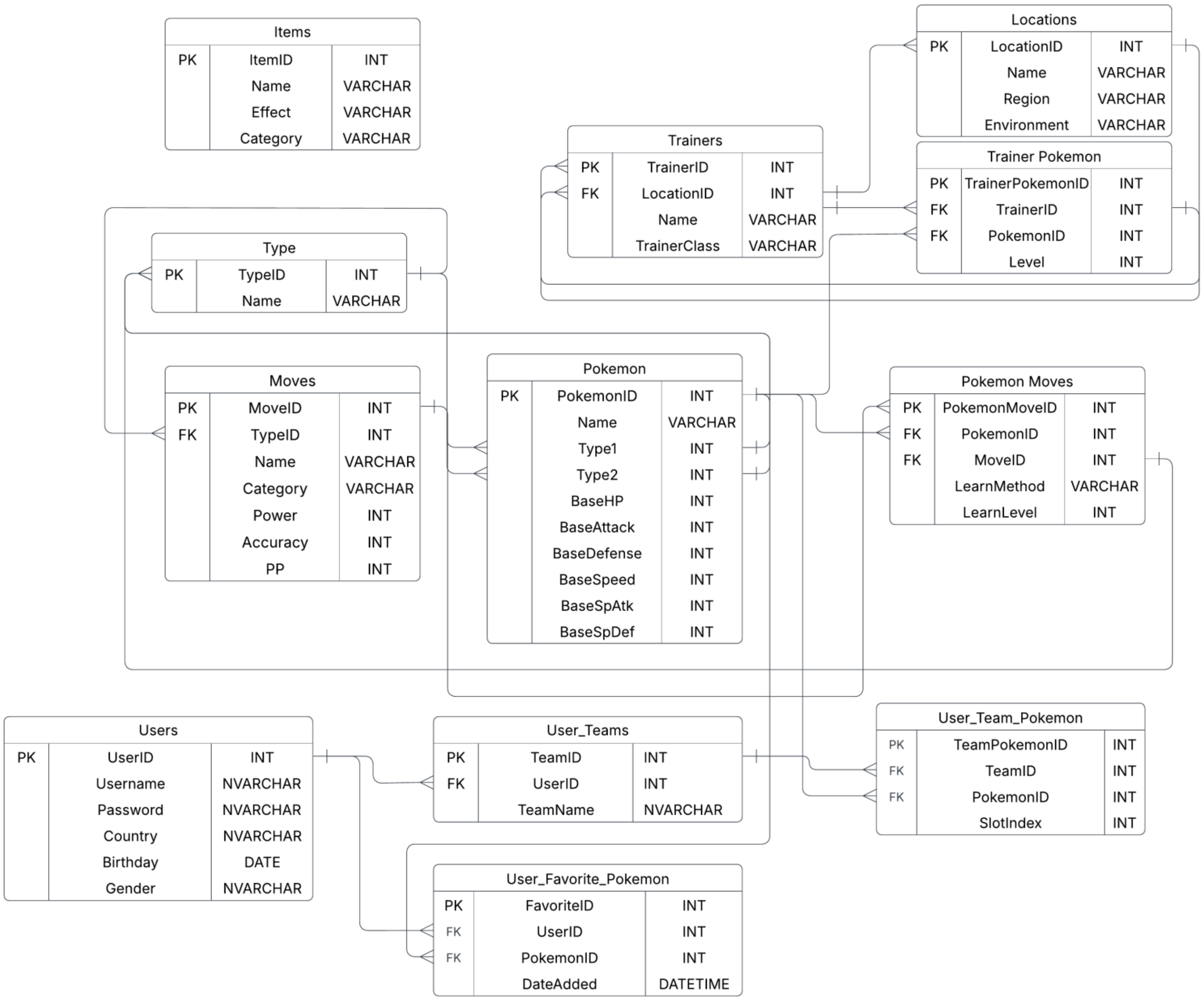


Figure 2: Entity Relationship Diagram

2. File Specification (Data Dictionary)

Database Name: PokéDexDB.DB

Table 1 Name: Items.TBL

Field Name	**Data Type**	**Length**	**Description**
ItemID	INT	N/A	Unique identifier for each item. Auto-incremented and set as the primary key.
Name	VARCHAR	100	Name of the item. Must be unique and cannot be null.
Effect	VARCHAR	MAX	Description of the item's effect. Cannot be null.
Category	VARCHAR	20	Category the item belongs to. Allowed values: 'PokéBall', 'Evolutionary', 'Key', 'Miscellaneous', 'Recovery', 'Battle', 'Fossil', 'Berry', 'Hold'.

Table 2: Users.TBL

Field Name	**Data Type**	**Length**	**Description**
UserID	INT	N/A	Unique identifier for each user. Auto-incremented and the primary key.
Username	NVARCHAR	50	The user's username. Must be unique and cannot be null.
Password	NVARCHAR	255	User password (stored securely). Cannot be null.
Country	NVARCHAR	50	Country of the user. Cannot be null.
Birthday	DATE	N/A	The user's date of birth. Cannot be null.
Gender	NVARCHAR	10	The user's gender. Allowed values: 'Male' or 'Female'. Cannot be null.

Table 3: User_Teams.TBL

Field Name	Data Type	Length	Description
TeamID	INT	N/A	Unique identifier for each team. Auto-incremented and the primary key.
UserID	INT	N/A	Foreign key referencing Users(UserID).
TeamName	NVARCHAR	50	Name of the team. Cannot be null.

Table 4: User_Team_Pokemon.TBL

Field Name	Data Type	Length	Description
TeamPokemonID	INT	N/A	Unique identifier for each Pokemon within a user team. Auto-incremented and the primary key.
TeamID	INT	N/A	Foreign key referencing User_Teams(TeamID).
SlotIndex	INT	N/A	Position of the Pokemon within the team. Must be between 0 and 5.
PokemonID	INT	N/A	Foreign key referencing Pokemon(PokemonID).

Table 5: User_Favorite_Pokemon.TBL

Field Name	Data Type	Length	Description
FavoriteID	INT	N/A	Unique identifier for each favorite entry. Auto-incremented and the primary key.
UserID	INT	N/A	Foreign key referencing Users(UserID).
PokemonID	INT	N/A	Foreign key referencing Pokemon(PokemonID).
DateAdded	DATETIME	N/A	Timestamp when the Pokemon was added to favorites. Defaults to the current date/time.

Table 6: Type.TBL

Field Name	**Data Type**	**Length**	**Description**
TypeID	INT	N/A	Unique identifier for each type. Set as the primary key.
Name	VARCHAR	20	Name of the type (e.g., Fire, Water, Grass). Cannot be null.

Table 7: Trainers.TBL

Field Name	**Data Type**	**Length**	**Description**
TrainerID	INT	N/A	Unique identifier for each trainer. Auto-incremented and the primary key.
Name	VARCHAR	255	Name of the trainer. Cannot be null.
TrainerClass	VARCHAR	50	The trainer's class. Allowed values: 'Trainer', 'Gym Leader', 'Elite Four', 'Champion'. Cannot be null.
LocationID	INT	N/A	Foreign key referencing Locations(LocationID).

Table 8: Trainer_Pokemon.TBL

Field Name	**Data Type**	**Length**	**Description**
TrainerPokemonID	INT	N/A	Unique identifier for each Trainer-Pokemon association. Set as the primary key.
TrainerID	INT	N/A	Foreign key referencing Trainers(TrainerID).
PokemonID	INT	N/A	Foreign key referencing Pokemon(PokemonID).
Level	INT	N/A	Level of the Pokemon when associated with the trainer. Cannot be null.

Table 9: Pokemon.TBL

Field Name	**Data Type**	**Length**	**Description**
PokemonID	INT	N/A	Unique identifier for each Pokemon. Set as the primary key.
Name	VARCHAR	50	Name of the Pokemon. Cannot be null.
Type1	INT	N/A	Foreign key referencing the primary type (Type) of the Pokemon.
Type2	INT	N/A	Secondary type of the Pokemon. Defaults to 0 if not applicable.
BaseHP	INT	N/A	Base hit points (HP) of the Pokemon.
BaseAttack	INT	N/A	Base attack stat of the Pokemon.
BaseDefense	INT	N/A	Base defense stat of the Pokemon.
BaseSpeed	INT	N/A	Base speed stat of the Pokemon.
BaseSpAtk	INT	N/A	Base special attack stat of the Pokemon.
BaseSpDef	INT	N/A	Base special defense stat of the Pokemon.

Table 10: Pokemon_Moves.TBL

Field Name	**Data Type**	**Length**	**Description**
PokemonMoveID	INT	N/A	Unique identifier for each Pokemon move entry. Auto-incremented and the primary key.
PokemonID	INT	N/A	Foreign key referencing Pokemon(PokemonID).
MoveID	INT	N/A	Foreign key referencing Moves(MoveID).
LearnMethod	VARCHAR	20	Method by which the Pokemon learns the move. Allowed values: 'Level-Up', 'TM', 'HM', 'Pre-Evolution'. May be null.
LearnLevel	INT	N/A	The level at which the move is learned (if applicable). May be null.

Table 11: Moves.TBL

Field Name	**Data Type**	**Length**	**Description**
MoveID	INT	N/A	Unique identifier for each move. Set as the primary key.
Name	VARCHAR	50	Name of the move. Cannot be null.
Category	VARCHAR	20	Category of the move. Cannot be null.
TypeID	INT	N/A	Foreign key referencing Type(TypeID) that represents the move's type.
Power	INT	N/A	The power of the move.
Accuracy	INT	N/A	The accuracy of the move.
PP	INT	N/A	The number of Power Points (PP) for the move.

Table 12: Locations.TBL

Field Name	**Data Type**	**Length**	**Description**
LocationID	INT	N/A	Unique identifier for each location. Set as the primary key.
Name	VARCHAR	50	Name of the location. Cannot be null.
Region	VARCHAR	50	Region where the location is found. Cannot be null.
Environment	VARCHAR	50	Type of environment (e.g., forest, cave, city). Cannot be null.

3. H-Diagram

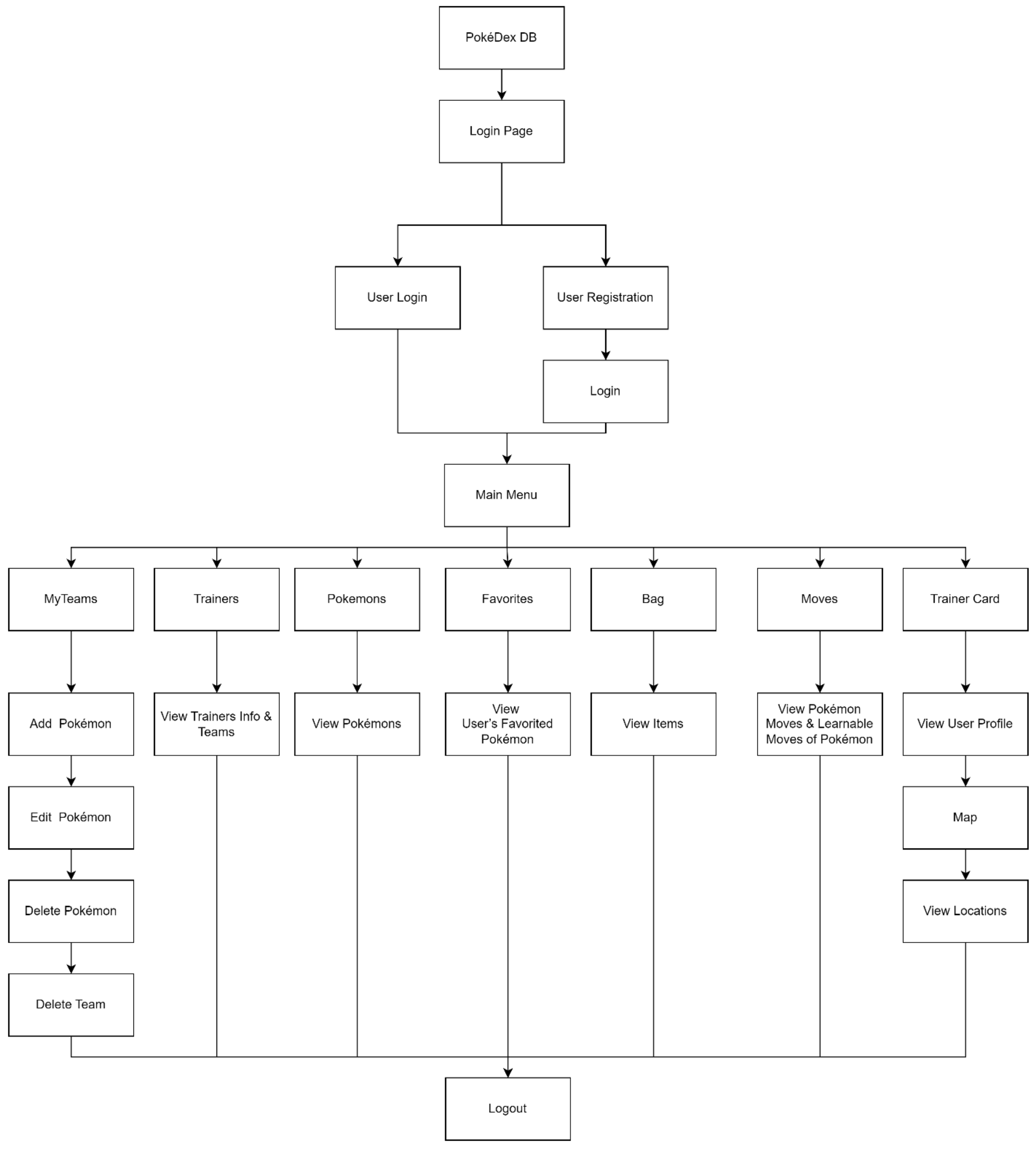


Figure 3 : H-Diagram for PokéDex DB

4. Modules and Functions

Authentication Module. This module manages user access and security by guiding users through the login and registration process. It begins with the LoginForm.cs where users can either enter their credentials to log in or choose to create a new account. New users are then led through a two-step registration process via the UserRegistrationForm.cs and UsernamePasswordForm.cs before returning to the login screen for authentication.

Main Menu Module. The Main Menu Module serves as the central hub for navigation, providing access to all primary features of the application. Once authenticated, users are directed to MainMenu.cs where they can easily select any of the available functions. This module ensures a seamless transition between different parts of the application.

Team Management Module. This module empowers users to create and manage their custom Pokémon teams. Through MyTeamsForm.cs, users can add, edit, and delete Pokémon to personalize their teams. The interface is designed to be intuitive, allowing for quick modifications to enhance the user experience.

Trainers Module. The Trainers Module displays comprehensive information about all in-game trainers, including Gym Leaders, Elite Four members, and Champions. It presents detailed profiles of each trainer along with their Pokémon teams via TrainersForm.cs. Users can explore trainer data to better understand the challenges and strategies within the game.

Pokémon Data Module. This module offers a rich view of the Pokémon database through PokemonForm.cs, featuring a DataGridView that displays animated GIFs, favorite markers, names, and detailed stats for each Pokémon. It enables users to browse and analyze Pokémon data efficiently. The design provides an engaging visual experience while delivering essential information about each Pokémon.

Favorites Module. The Favorites Module allows users to quickly access Pokémon they have marked as favorites. Through FavoritesForm.cs, users can view a curated list of their preferred Pokémon, making it easier to build strategies and manage teams. This module enhances the user experience by offering personalized content.

Items Module. This module manages the in-game item inventory via BagForm.cs, displaying detailed information about each item. Users can search for items by name, view their effects, and understand their categories such as healing, evolution, or battle items. The Items Module ensures that players have easy access to essential game resources.

Moves Module. The Moves Module provides a comprehensive list of all Pokémon moves available in the game through MoveForm.cs. It displays critical details like power, accuracy, PP, and move category, along with the Pokémon that are eligible to learn each move. This module assists users in strategizing their moves.

User Profile Module. This module, accessed via TrainerCardForm.cs, displays the user’s personal profile including name, country, birthday, and gender. It serves as a centralized location for users to review their personal details. The User Profile Module reinforces the connection between the user and the game by personalizing the experience.

Map Module. Embedded within the User Profile Module, the Map Module provides interactive maps of the game world through MapForm.cs. Users can click on specific map areas to retrieve detailed location information and view trainers present in those regions. This module enhances navigation and immersion within the game environment.

Logout Module. The Logout Module ensures that user sessions are terminated securely to prevent unauthorized access. It handles the process of logging out, closing the session, and redirecting the user appropriately. This module is vital for maintaining the security and integrity of user data within the application.

5. PROGRAM SPECIFICATION

Listing 1: Authentication Module Pseudocode

```
FUNCTION AuthenticationModule():
  DISPLAY "LoginForm: Please enter your username and password."
  userAction ← GET_USER_ACTION()  // Options: "Login" or "Create Account"
  IF userAction == "Login" THEN
    username ← INPUT("Enter Username:")
    password ← INPUT("Enter Password:")
    IF VALIDATE_CREDENTIALS(username, password) THEN
      CALL MainMenuModule()
    ELSE
      DISPLAY "Invalid credentials. Please try again."
      CALL AuthenticationModule()
    ENDIF
  ELSE IF userAction == "Create Account" THEN
    CALL UserRegistrationModule()
  ENDIF
END FUNCTION
```

Listing 2: User Registration Module Pseudocode

```
FUNCTION UserRegistrationModule():
    DISPLAY "UserRegistrationForm: Enter your country, birthday, and gender."
    country ← INPUT("Country:")
    birthday ← INPUT("Birthday:")
    gender ← INPUT("Gender:")
    DISPLAY "UsernamePasswordForm: Set your username and password."
    username ← INPUT("Username:")
    password ← INPUT("Password:")
    confirmPassword ← INPUT("Confirm Password:")
    IF password == confirmPassword THEN
        SAVE_NEW_USER(username, password, country, birthday, gender)
        DISPLAY "Registration successful. Please log in."
        CALL AuthenticationModule()
    ELSE
        DISPLAY "Passwords do not match. Please try again."
        CALL UserRegistrationModule()
    ENDIF
END FUNCTION
```

Listing 3: Main Menu Module Pseudocode

```
FUNCTION MainMenuModule():
    DISPLAY "Main Menu: Choose an option."
    DISPLAY "1. Team Management"
    DISPLAY "2. Trainers Info"
    DISPLAY "3. Pokémon Data"
    DISPLAY "4. Favorites"
    DISPLAY "5. Items"
    DISPLAY "6. Moves"
    DISPLAY "7. User Profile"
    DISPLAY "8. Logout"
    option ← INPUT("Select an option:")
    SWITCH option:
        CASE "1": CALL TeamManagementModule()
        CASE "2": CALL TrainersModule()
        CASE "3": CALL PokemonDataModule()
        CASE "4": CALL FavoritesModule()
        CASE "5": CALL ItemsModule()
        CASE "6": CALL MovesModule()
        CASE "7": CALL UserProfileModule()
        CASE "8": CALL LogoutModule()
        DEFAULT:
            DISPLAY "Invalid option. Try again."
            CALL MainMenuModule()
    END SWITCH
END FUNCTION
```

Listing 4: Team Management Module Pseudocode

```
FUNCTION TeamManagementModule():
    DISPLAY "Team Management: Options - 1. Add Pokémon  2. Edit Pokémon  3. Delete Pokémon"
    choice ← INPUT("Select an option:")
    IF choice == "1" THEN
        CALL AddPokemonModule()
```

```
  ELSE IF choice == "2" THEN
    CALL EditPokemonModule()
  ELSE IF choice == "3" THEN
    CALL DeletePokemonModule()
  ELSE
    DISPLAY "Invalid option."
  ENDIF
  CALL MainMenuModule()
END FUNCTION
```

Listing 5: Add Pokémon Module Pseudocode

```
FUNCTION AddPokemonModule():
  DISPLAY "Select a Pokémon from the Pokédex to add to your team."
  selectedPokemon ← INPUT("Enter Pokémon ID or Name:")
  ADD_POKEMON_TO_TEAM(selectedPokemon)
  DISPLAY "Pokémon added successfully."
END FUNCTION
```

Listing 6: Edit Pokémon Module Pseudocode

```
FUNCTION EditPokemonModule():
  DISPLAY "Your Current Team: [List of Pokémon]"
  selectedTeam ← INPUT("Select a Team to edit:")
  newDetails ← INPUT("Enter new Details:")
  UPDATE_TEAM_POKEMON(selectedPokemon, newDetails)
  DISPLAY "Pokémon details updated."
END FUNCTION
```

Listing 7: Delete Pokémon Module Pseudocode

```
FUNCTION DeletePokemonModule():
    DISPLAY "Your Current Team: [List of Pokémon]"
    selectedPokemon ← INPUT("Select a Pokémon to delete:")
    REMOVE_POKEMON_FROM_TEAM(selectedPokemon)
    DISPLAY "Pokémon deleted."
END FUNCTION
```

Listing 8: Trainers Module Pseudocode

```
FUNCTION TrainersModule():
    DISPLAY "Trainers Information: Displaying all trainer profiles and their teams."
    SHOW_TRAINERS_DATA()
    CALL MainMenuModule()
END FUNCTION
```

Listing 9: Pokémon Data Module Pseudocode

```
FUNCTION PokemonDataModule():
    DISPLAY "Pokémon Data: Viewing Pokémon with GIFs, favorites, names, and stats."
    SHOW_POKEMON_DATA_GRID()
    CALL MainMenuModule()
END FUNCTION
```

Listing 10: Favorites Module Pseudocode

```
FUNCTION FavoritesModule():
    DISPLAY "Favorites: Listing all favorited Pokémon."
    SHOW_FAVORITE_POKEMON()
    CALL MainMenuModule()
END FUNCTION
```

Listing 11: Items Module Pseudocode

```
FUNCTION ItemsModule():
    DISPLAY "Items Inventory: Listing all in-game items and details."
    SHOW_ITEMS_DATA()
    CALL MainMenuModule()
END FUNCTION
```

Listing 12: Moves Module Pseudocode

```
FUNCTION MovesModule():
    DISPLAY "Moves Information: Listing all moves and the Pokémon eligible to learn them."
    SHOW_MOVES_DATA()
    CALL MainMenuModule()
END FUNCTION
```

Listing 13: User Profile Module Pseudocode

```
FUNCTION UserProfileModule():
    DISPLAY "User Profile: Displaying your personal details (name, country, birthday, gender)."
    SHOW_USER_PROFILE()
    userChoice ← INPUT("Press 'M' to view Map or any other key to return:")
    IF userChoice == "M" THEN
```

```
        CALL MapModule()
    ELSE
        CALL MainMenuModule()
    ENDIF
END FUNCTION
```

Listing 14: Map Module Pseudocode

```
FUNCTION MapModule():
    DISPLAY "Interactive Map: Click on areas to view location details and associated trainers."
    selectedArea ← INPUT("Select a map area:")
    SHOW_MAP_AREA_INFO(selectedArea)
    CALL MainMenuModule()
END FUNCTION
```

Listing 15: Logout Module Pseudocode

```
FUNCTION LogoutModule():
    DISPLAY "Logging out..."
    TERMINATE_SESSION()
    CALL AuthenticationModule()
END FUNCTION
```

Program Entry Point

```
BEGIN PROGRAM
    CALL AuthenticationModule()
END PROGRAM
```

6. Sample Screenshots

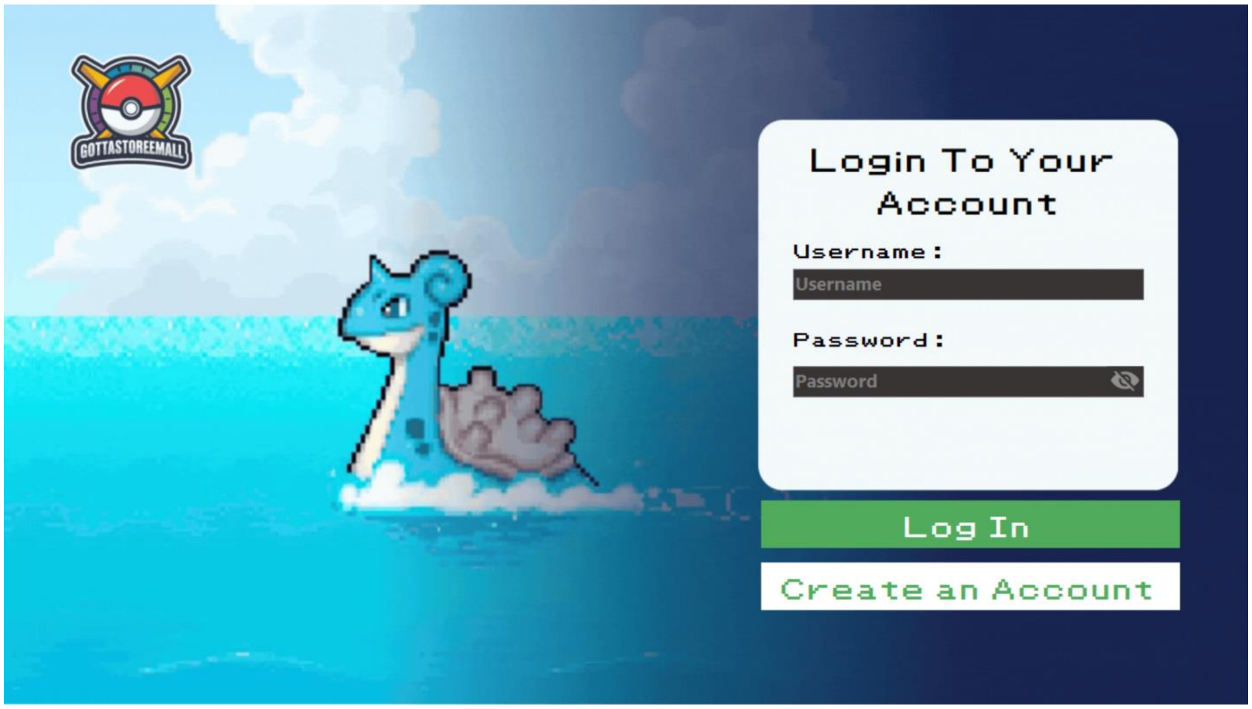
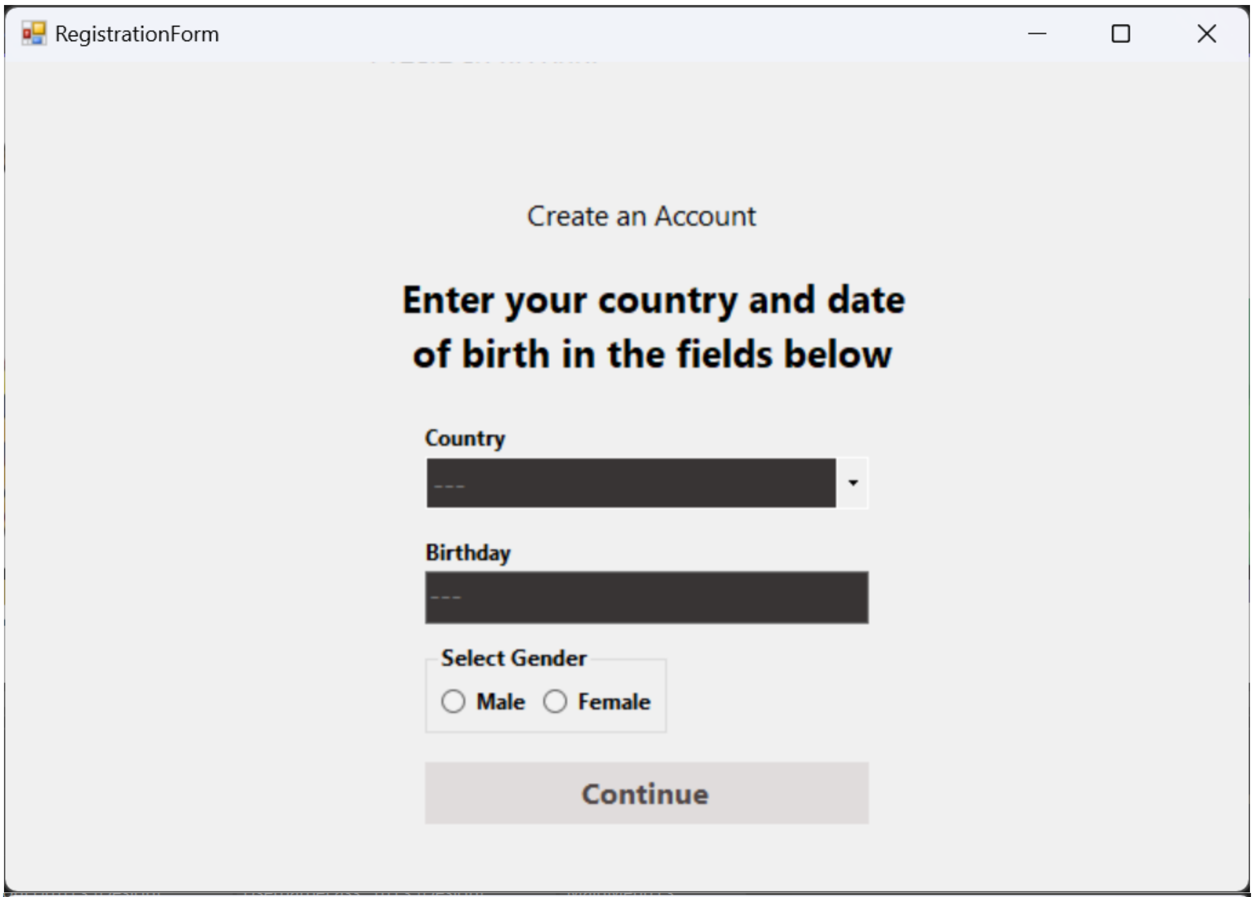


Figure 4: Login


RegistrationForm

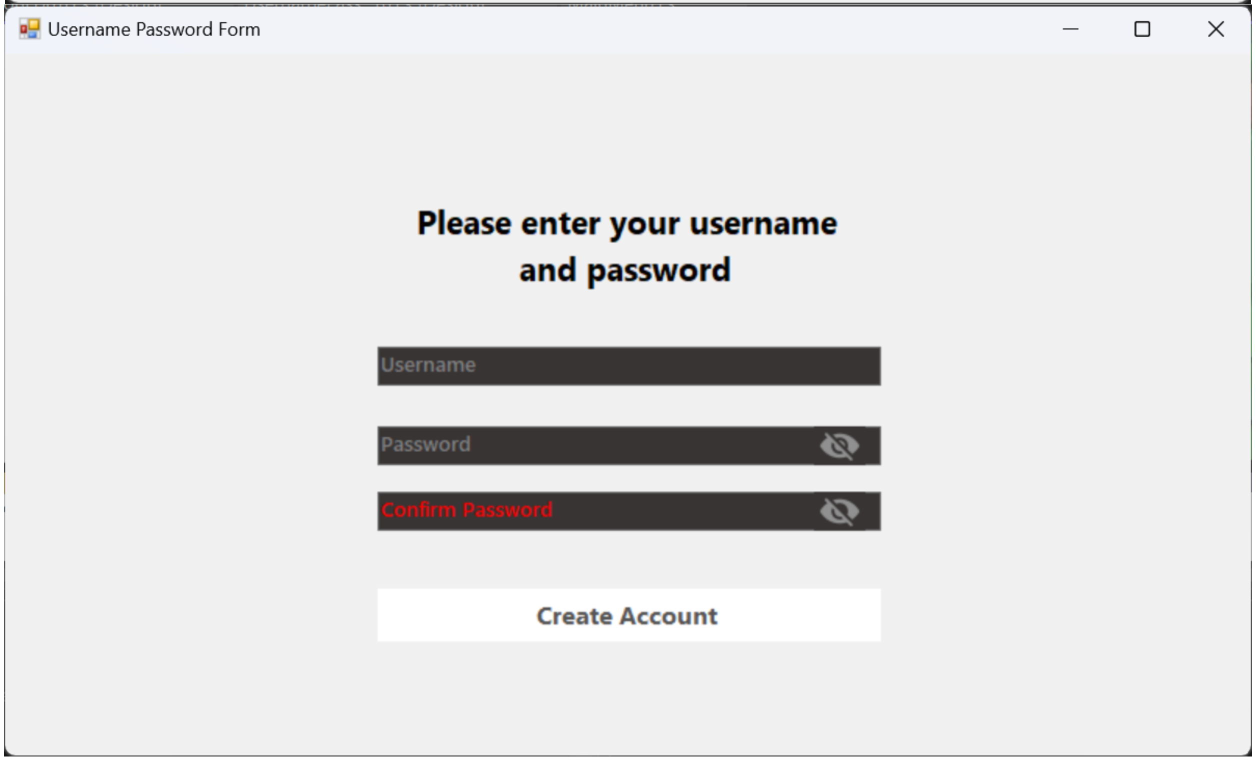
Create an Account

Enter your country and date of birth in the fields below

Country: ---

Birthday: ---

Select Gender: ○ Male ○ Female


Username Password Form

Please enter your username and password

Username

Password

Confirm Password

Figure 5: Registration

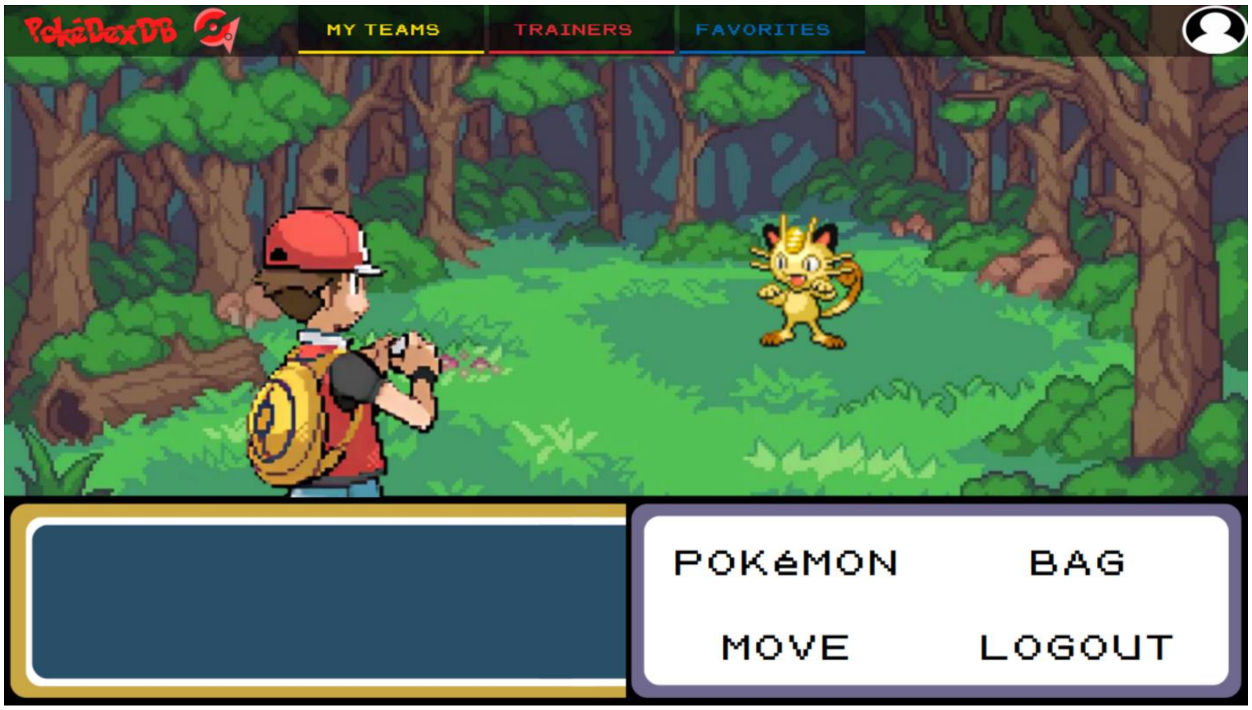


Figure 6: Registration

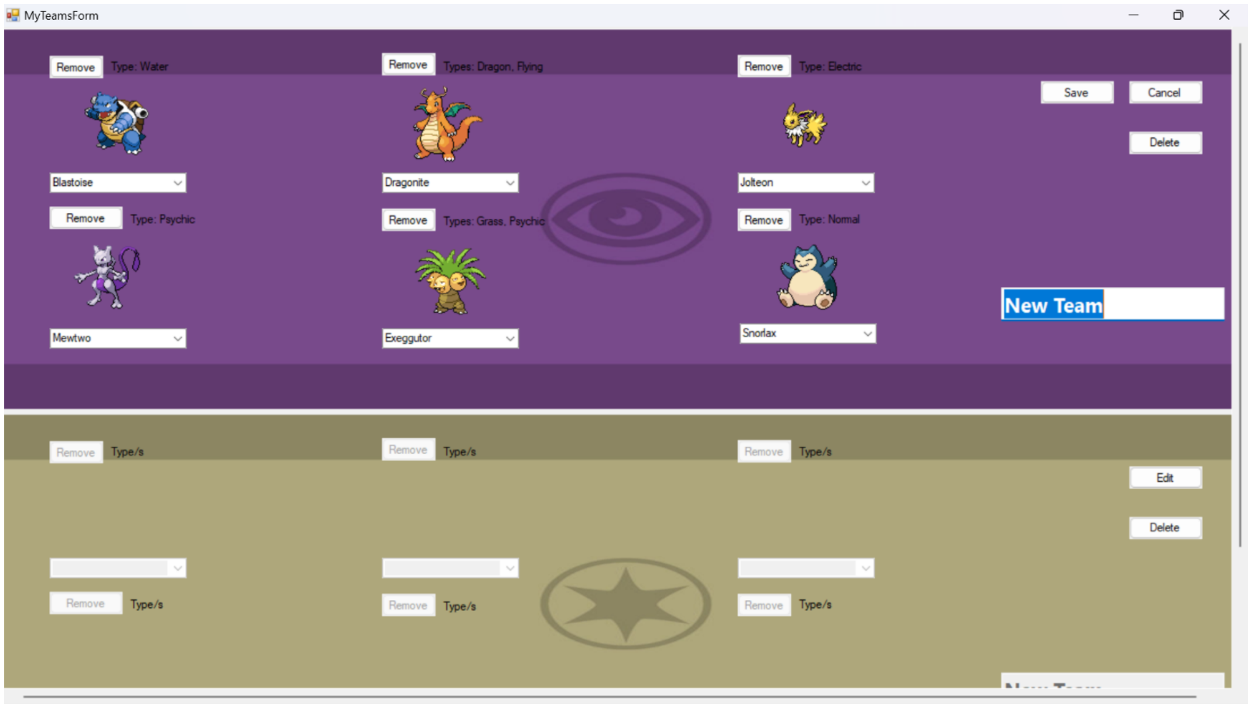
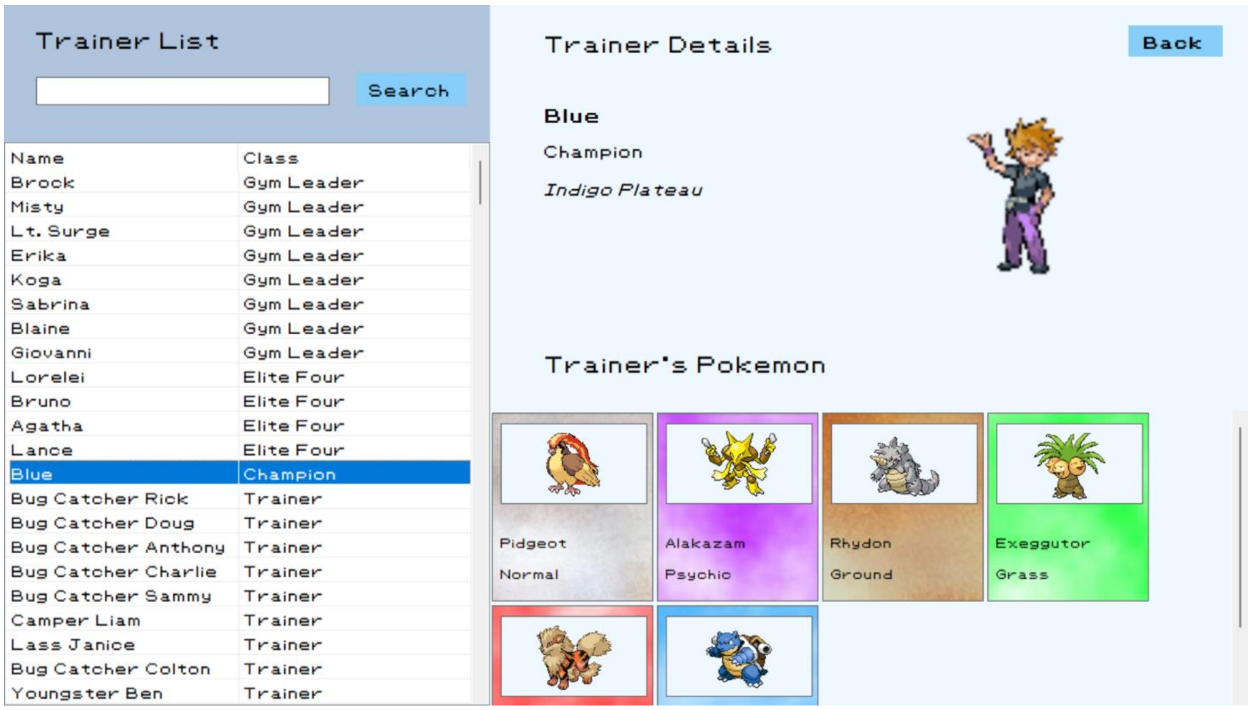


Figure 7: My Teams

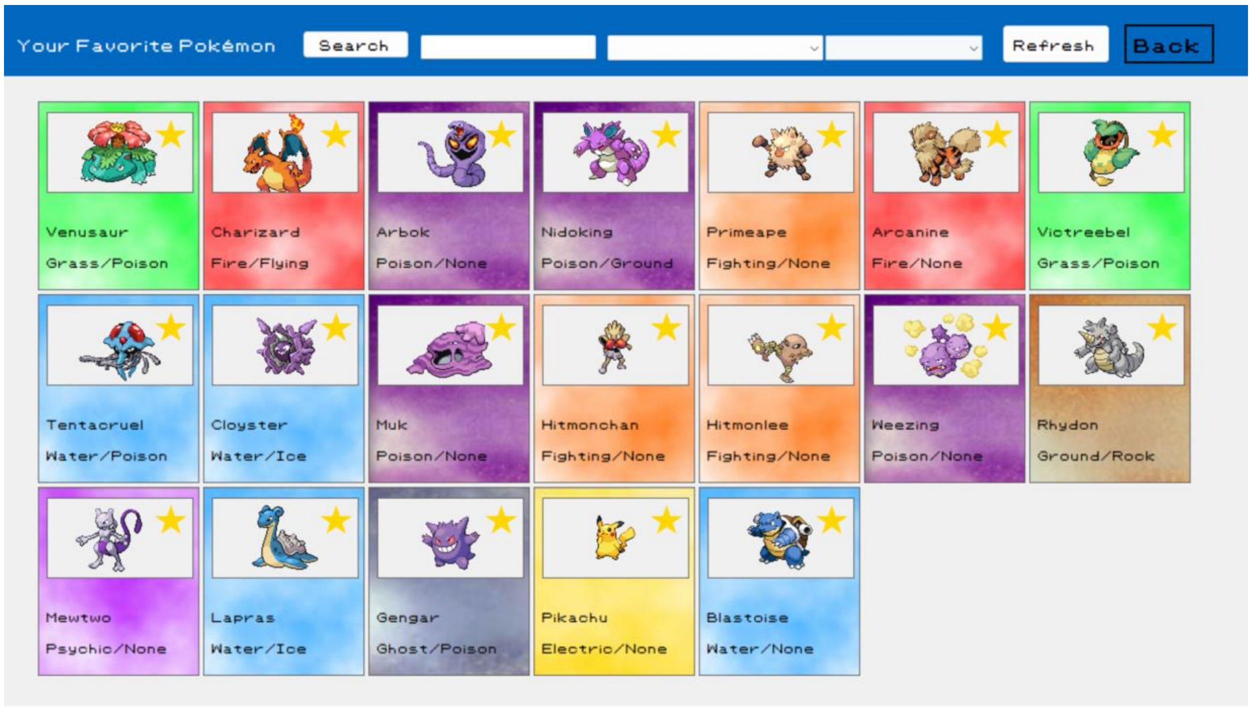


Name	Class
Brock	Gym Leader
Misty	Gym Leader
Lt. Surge	Gym Leader
Erika	Gym Leader
Koga	Gym Leader
Sabrina	Gym Leader
Blaine	Gym Leader
Giovanni	Gym Leader
Lorelei	Elite Four
Bruno	Elite Four
Agatha	Elite Four
Lance	Elite Four
Blue	Champion
Bug Catcher Rick	Trainer
Bug Catcher Doug	Trainer
Bug Catcher Anthony	Trainer
Bug Catcher Charlie	Trainer
Bug Catcher Sammy	Trainer
Camper Liam	Trainer
Lass Janice	Trainer
Bug Catcher Colton	Trainer
Youngster Ben	Trainer

Blue
Champion
Indigo Plateau

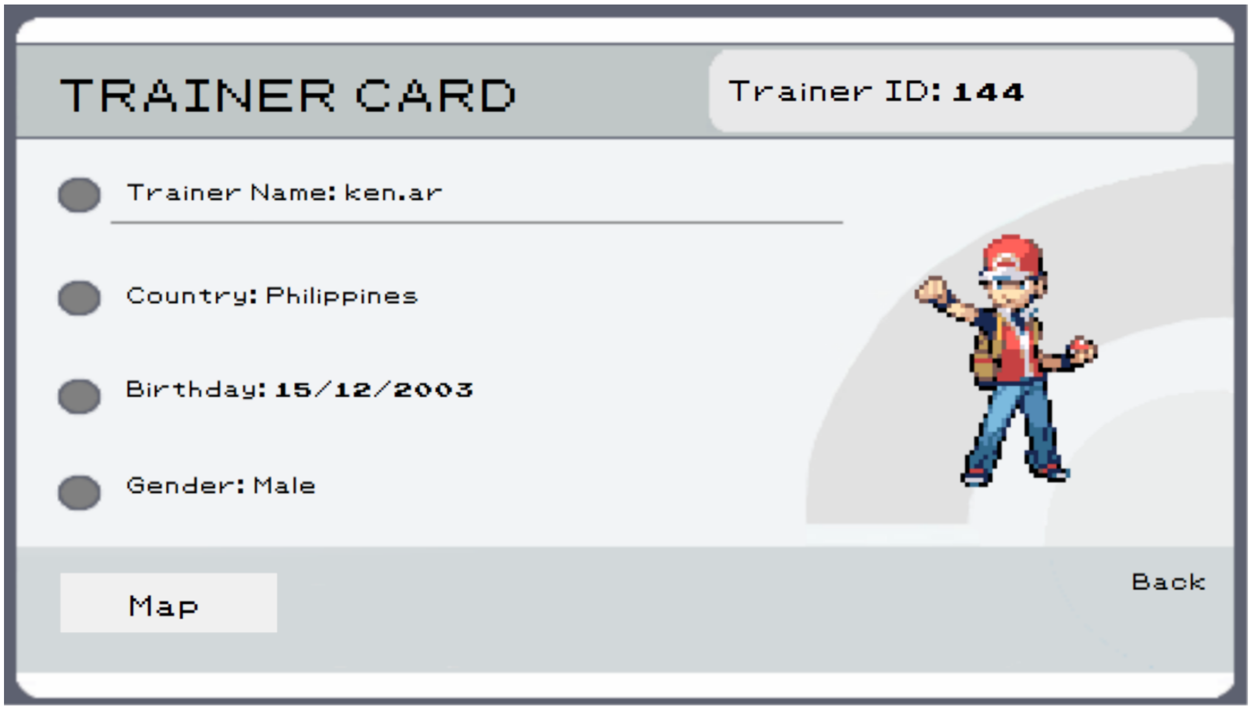
Pidgeot
Normal
Alakazam
Psychic
Rhydon
Ground
Exeggutor
Grass

Figure 8: Trainers



Venusaur
Grass/Poison
Charizard
Fire/Flying
Arbok
Poison/None
Nidoking
Poison/Ground
Primeape
Fighting/None
Arcanine
Fire/None
Victreebel
Grass/Poison
Tentacruel
Water/Poison
Cloyster
Water/Ice
Muk
Poison/None
Hitmonchan
Fighting/None
Hitmonlee
Fighting/None
Weezing
Poison/None
Rhydon
Ground/Rock
Mewtwo
Psychic/None
Lapras
Water/Ice
Gengar
Ghost/Poison
Pikachu
Electric/None
Blastoise
Water/None

Figure 9: Favorite



TRAINER CARD

Trainer ID: 144

Trainer Name: ken.ar

Country: Philippines

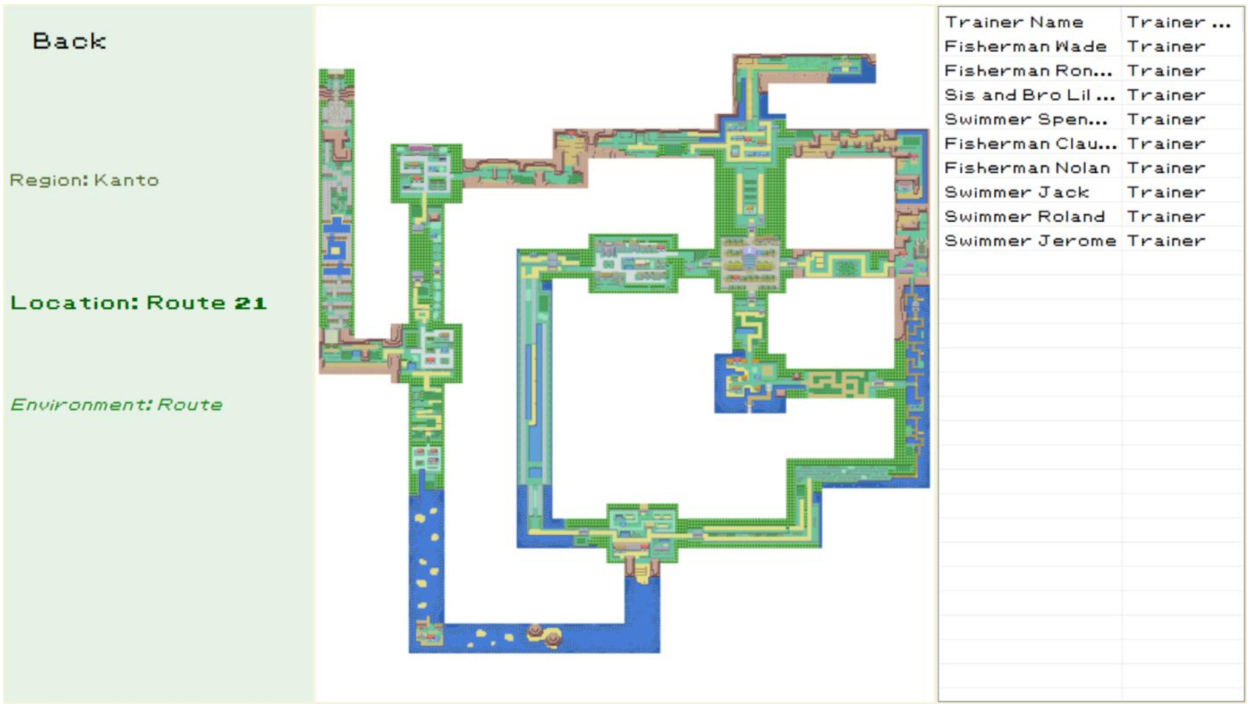
Birthday: 15/12/2003

Gender: Male

Map

Back

Figure 10: User Profile



Back

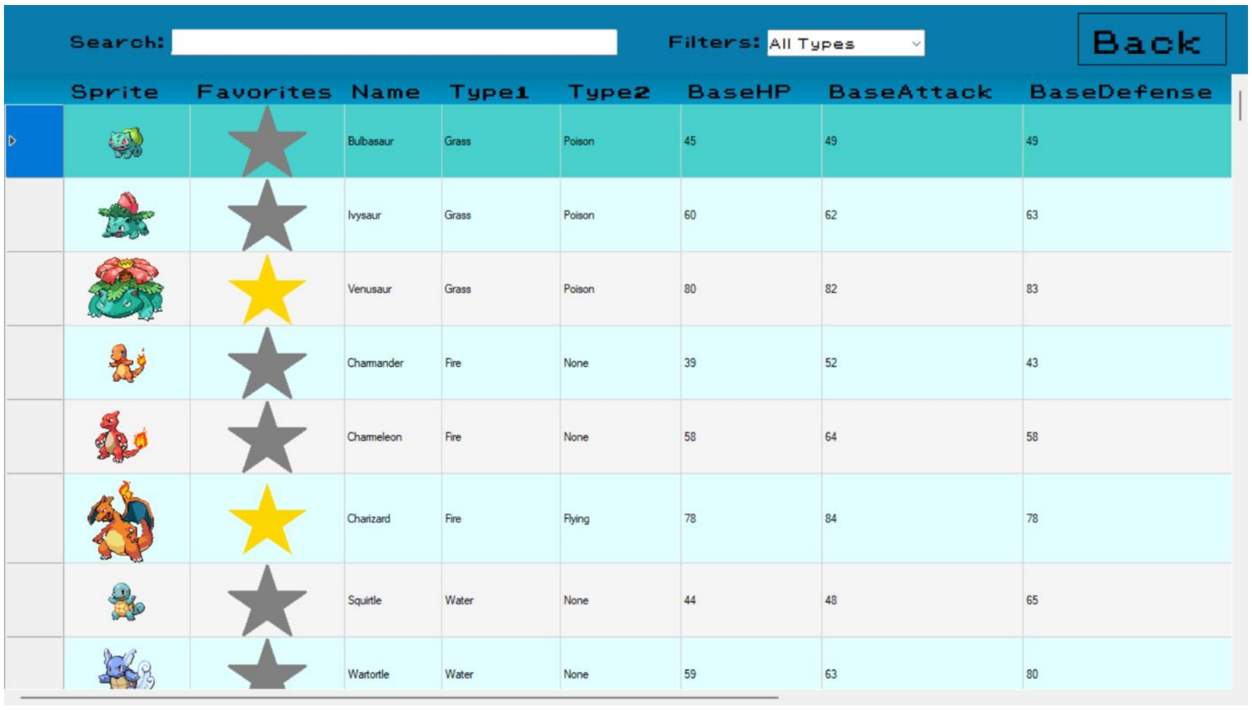
Region: Kanto

Location: Route 21

Environment: Route

Trainer Name	Trainer ...
Fisherman Wade	Trainer
Fisherman Ron...	Trainer
Sis and Bro Lil ...	Trainer
Swimmer Spen...	Trainer
Fisherman Clau...	Trainer
Fisherman Nolan	Trainer
Swimmer Jack	Trainer
Swimmer Roland	Trainer
Swimmer Jerome	Trainer

Figure 11: Map



Sprite	Favorites	Name	Type1	Type2	BaseHP	BaseAttack	BaseDefense
	☆	Bulbasaur	Grass	Poison	45	49	49
	☆	Ivysaur	Grass	Poison	60	62	63
	★	Venusaur	Grass	Poison	80	82	83
	☆	Charmander	Fire	None	39	52	43
	☆	Charmeleon	Fire	None	58	64	58
	★	Charizard	Fire	Flying	78	84	78
	☆	Squirtle	Water	None	44	48	65
	☆	Wartortle	Water	None	59	63	80

Figure 12: Pokemon

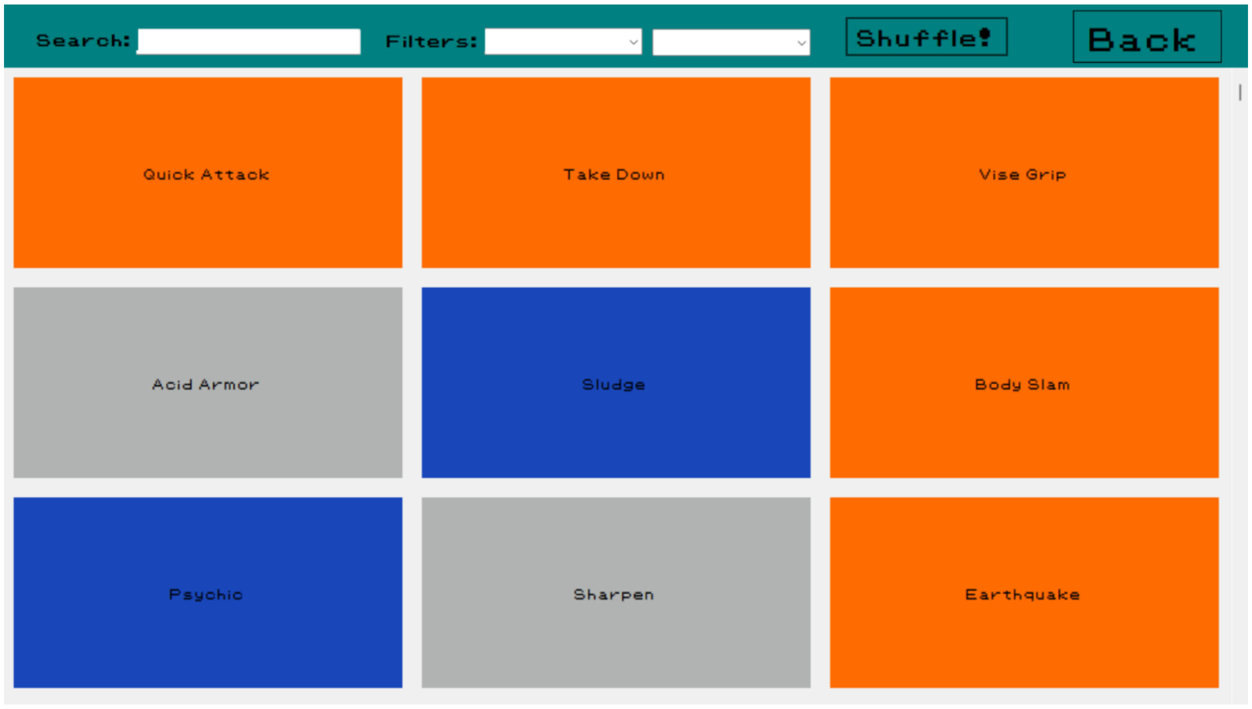
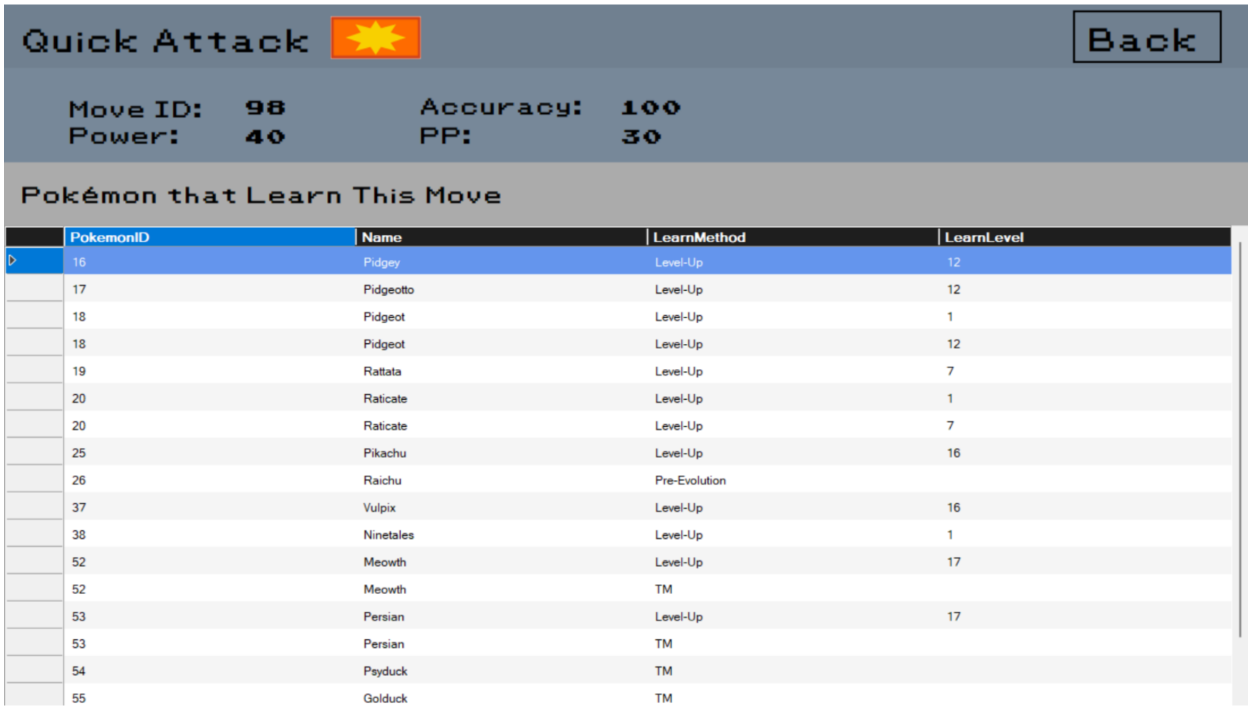


Figure 13: Moves

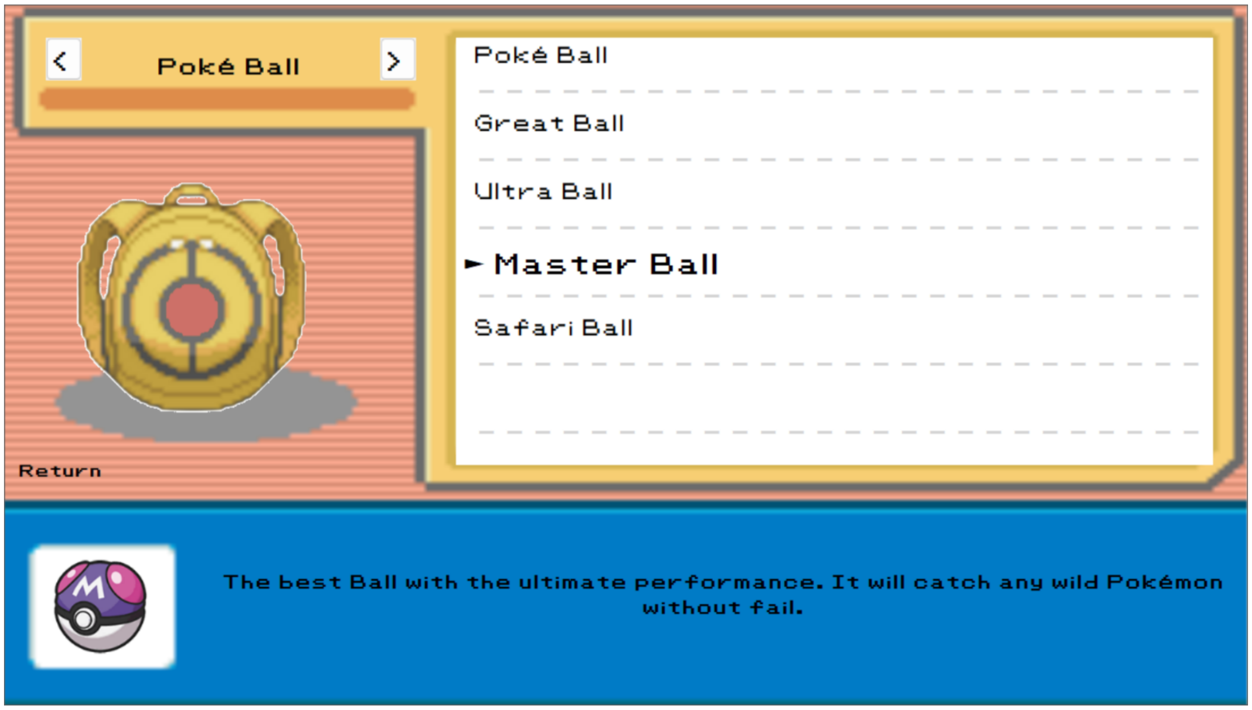


Move ID: 98
Power: 40
Accuracy: 100
PP: 30

Pokémon that Learn This Move

PokemonID	Name	LearnMethod	LearnLevel
16	Pidgey	Level-Up	12
17	Pidgeotto	Level-Up	12
18	Pidgeot	Level-Up	1
18	Pidgeot	Level-Up	12
19	Rattata	Level-Up	7
20	Raticate	Level-Up	1
20	Raticate	Level-Up	7
25	Pikachu	Level-Up	16
26	Raichu	Pre-Evolution	
37	Vulpix	Level-Up	16
38	Ninetales	Level-Up	1
52	Meowth	Level-Up	17
52	Meowth	TM	
53	Persian	Level-Up	17
53	Persian	TM	
54	Psyduck	TM	
55	Golduck	TM	

Figure 14: Moves Details



The best Ball with the ultimate performance. It will catch any wild Pokémon without fail.

Figure 15: Bag

Improvements:

Schema Optimization:

Refine database normalization and add indexing to enhance performance and ensure data consistency.

Enhanced Security:

Implement stronger authentication measures (e.g., multi-factor authentication) and data encryption to protect user information.

Improved User Interface:

Streamline form layouts and navigation to create a more intuitive and responsive front-end experience.

Expanded Data Insights:

Increase dataset volume and develop complex multi-table queries to derive deeper analytical insights.

Advanced Reporting:

Integrate interactive dashboards and real-time visualizations (using tools like Power BI and Python) to improve data-driven decision-making.