Game Dev project design



Members

Harry Tenenbaum (Lead) | Matthew Young Thomas Fisher | Andrew Coggin

Initial Application Design

Gold Team

# Outline

* Project Summary
* Requirements & Assumptions
* Business Process Overview
  1. High Level
  2. Low Level
* Data Design
  1. Models
  2. Dictionary
  3. Files
* Application Design
  1. Wireframes
  2. Context Diagram
  3. Data Flows
     1. Level 0
     2. 1.0
     3. 2.0
     4. 3.0
     5. 4.0
  4. Class Diagrams
     1. Hierarchy
     2. Blueprints
        1. GameObject
        2. Pet
        3. Ability
        4. Player

# Project Summary

The Gold Team has been contracted to develop a new game called BattlePets. This turn-based game allows the player to create and control a team of pets. The user will use their pet army to battle wild pets and other players’ pets to improve their own pets’ level and rarity, capture new pets, and battle other players’ pet teams. This project will give The Gold Team practical, firsthand experience in business analysis, application design, and an introduction to collaborative development. In addition to the design, the game will feature a database to store game data.

# Requirements & Assumptions

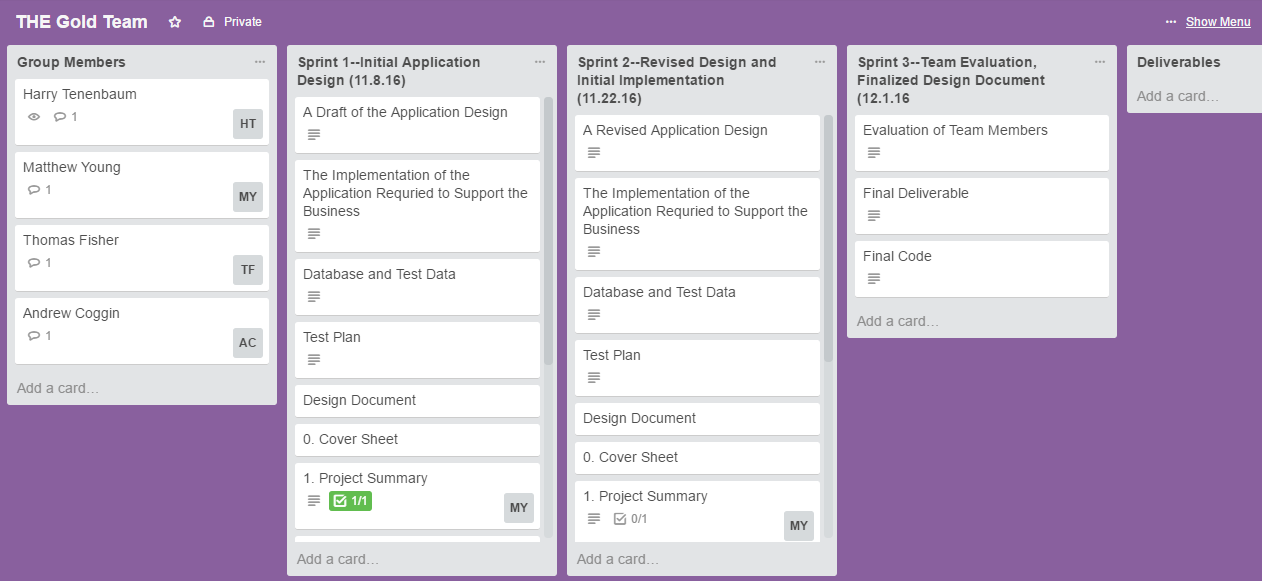
We negotiated the scope of the project with the client and determined that seeing only the pet that is battling on the battle screen was sufficient.

# Business Project Overview

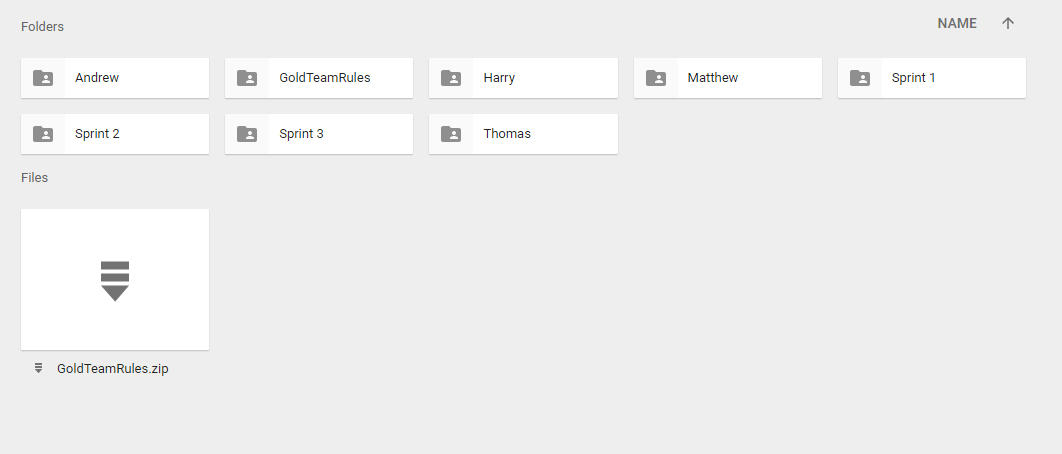
## High Level

By using multiple productivity resources, The Gold Team can communicate efficiently to effectively complete the deliverables for each sprint of the Game Design Project. Trello, Google Drive, and the “Update History” list in Microsoft Visual Studios serve as means for the team to keep track of and assign responsibilities, store files, and track updates of product development.

## Low Level

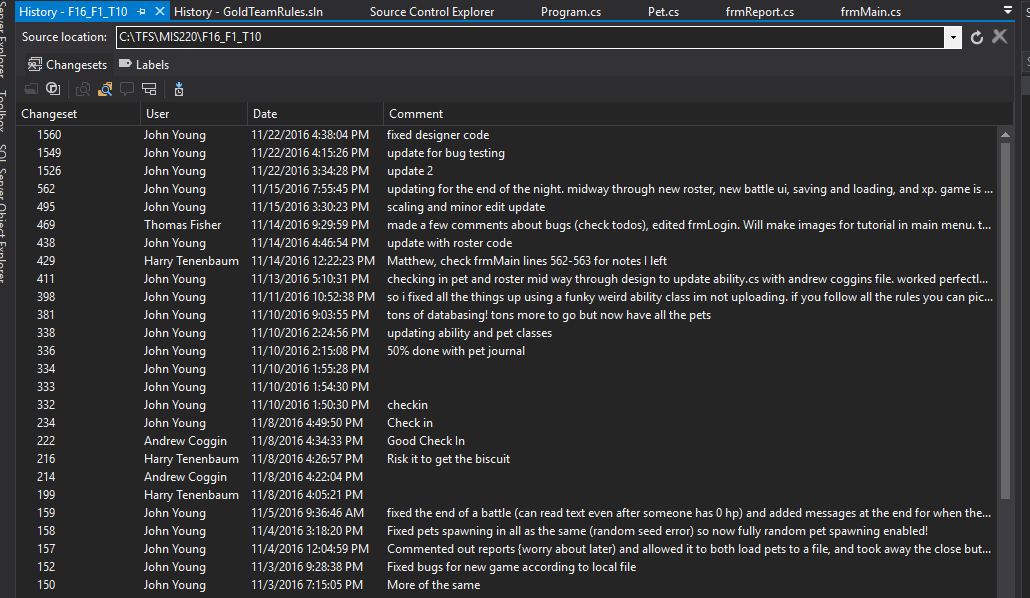
  
*The Gold Team Trello Page*

Our Trello page is designed to keep the group in check and ensure that it can complete its assignments. Each list (except for the “Group Members” and “Deliverables” lists) represents each sprint of the project. The lists contain a checklist with the deliverables to completed for the given sprint. Each deliverable then contains a more detailed checklist with the required components of the deliverable. The checklists help with the agile process, and allows each group member to be assigned to the various deliverables of the sprint.



*The Gold Team Google Drive*

The Google Drive folder contains each group member’s personal storage folder and the sprint folders. Included in each group member’s folder is their portion of the work for each sprint. Each sprint folder holds the document and associated code to be turned in. Also included in the Google Drive is a backup version of the application, updated periodically.

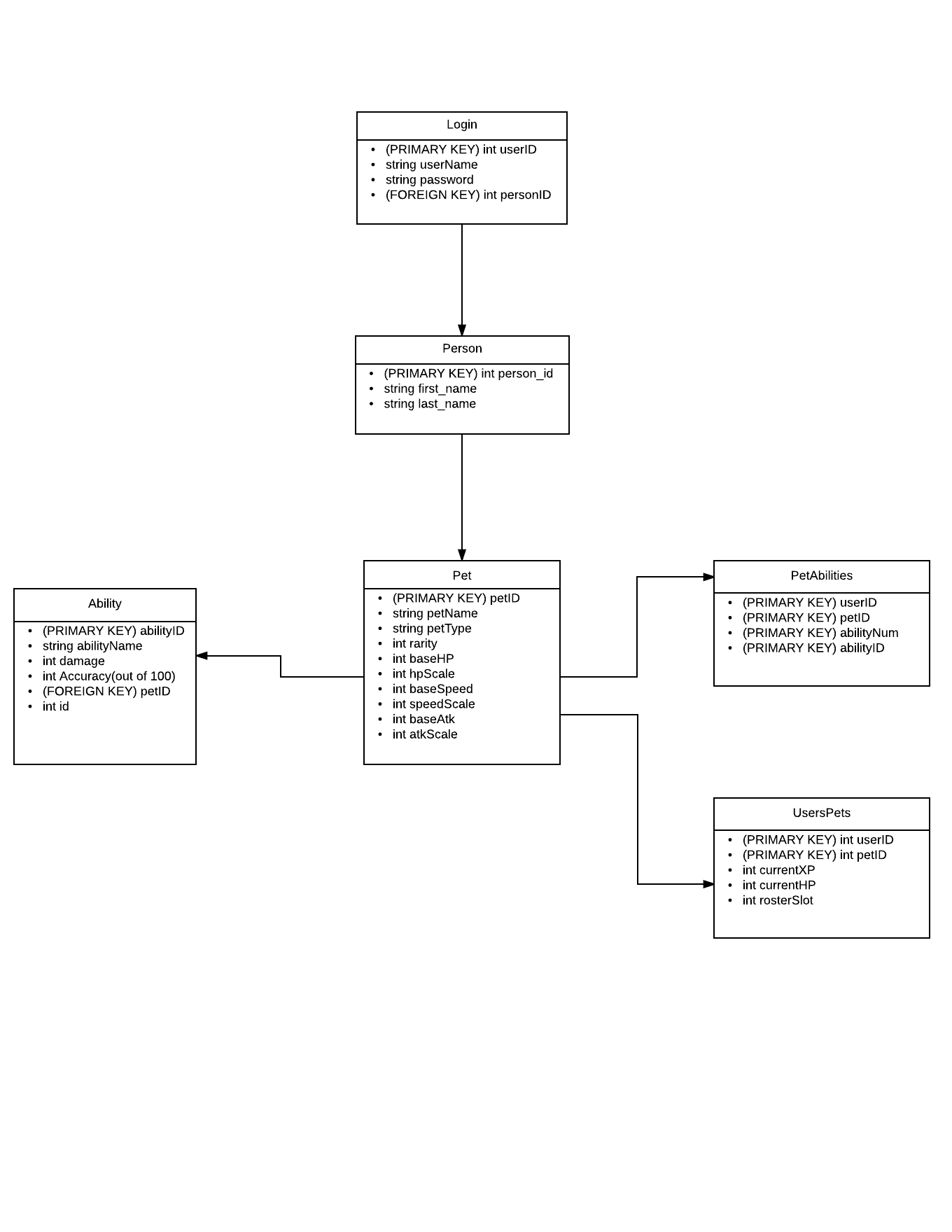


*The Gold Team Update History*

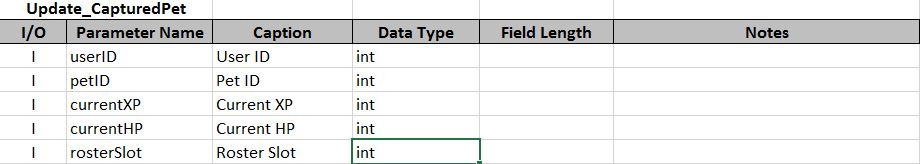
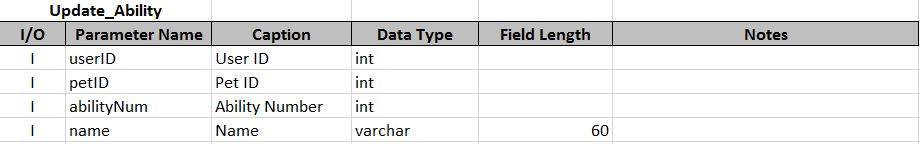
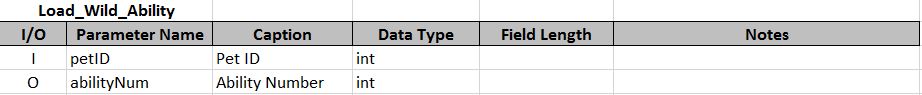
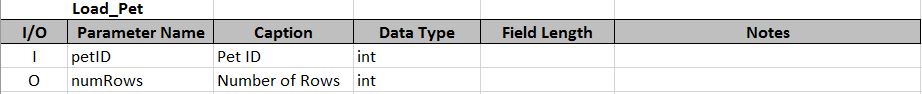
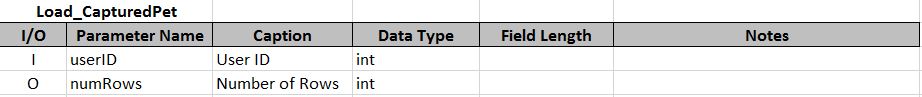
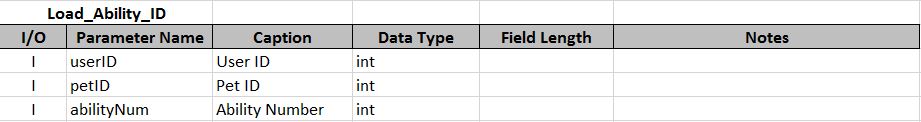
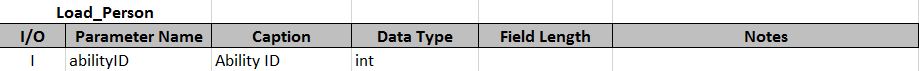
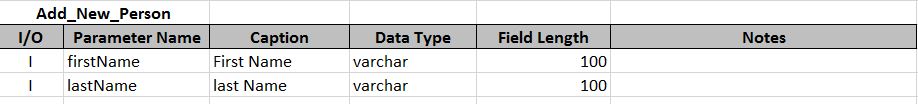
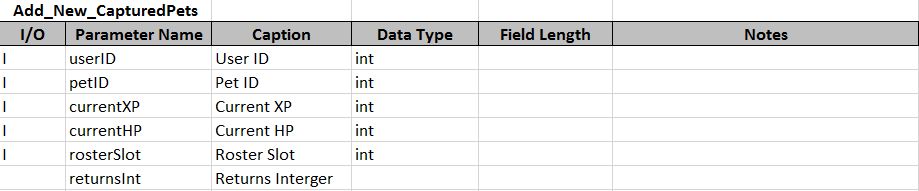
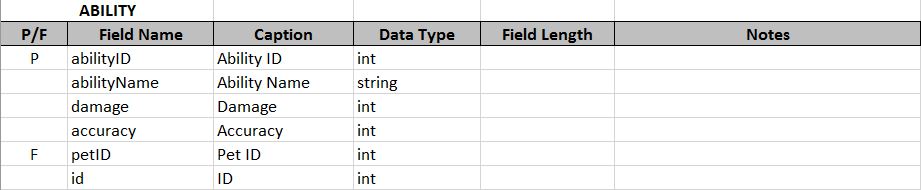
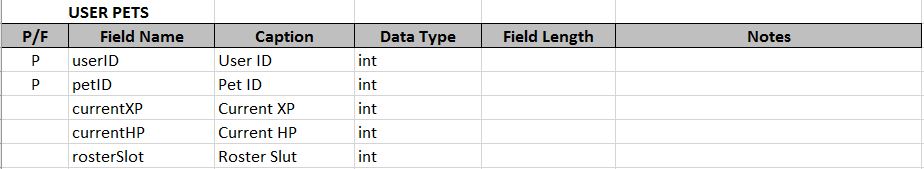
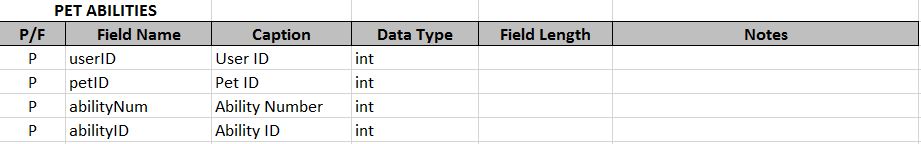
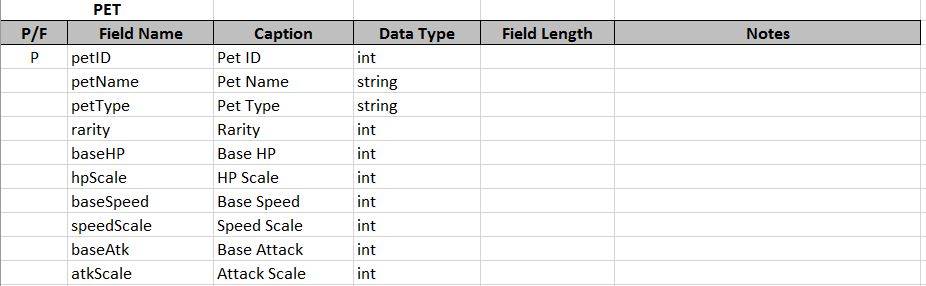
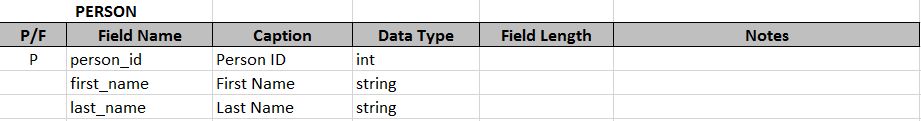
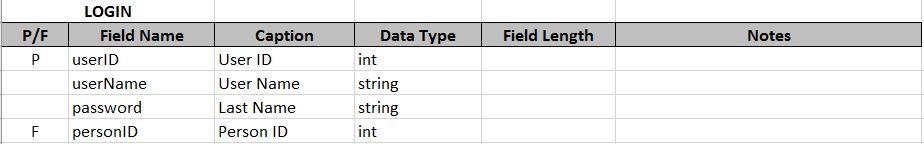
The history of the application keeps track of all updates made to the program and attached database. Comments attached to the frequent updates keep everyone on track with the progress of the applications. The comments also serve as a method for the group members to communicate with each other and leave messages about the development of the application.

# Data Design

## Models



## Dictionary



## Files

### Input

Resources:

* All images (i.e. background, sprites)

# Application Design

## Interactivity Diagram

Load Data

Save

GUI

Server

Calculate

Draw Screen

Save Data

Save

Receive Input

Player

## Wireframes

### Login

Welcome to Battle Pets!

Enter username here

Enter password here

Password

Username

Cancel

Login

Background Color

### Create Account

First Name

Enter first name here

Enter last name here

Last Name

Enter password here

Password

Create Account

Enter username here

Username

Cancel

Create

Background Color

### Starting Pet Screen

Pet 1 Info

Pet 2 Info

Pet 3 Info

Pet 2 Image

Pet 1 Image

Pet 3 Image

Done

Background Color

### MapBattle

Pet

Pet

Pet

Player

Background Image

Enemy Info

Player Info

Wild Pet

Your Pet

Menu Options

Background Color

### Forfeit Form

Background Color

Cancel

Yes

Are you sure?

Menu

Menu Options

Background Color

### Pet Journal

List of all pets

Pet Image

Pet Info

Cancel

Heal

Background Color

### Pet Roster

List of pets not in roster

Pet Image

Pet Info

Cancel

Choose

Background Color

### Report

Background Color

Battle #

Battle Report Info

## Context Diagrams

### Report Layout

ComboBox on top with Battle Number

Data Grid View with detail of battle

#### Columns:

Start:

Timestamp | BattleStarted | Pet 1 Info | Pet 2 Info

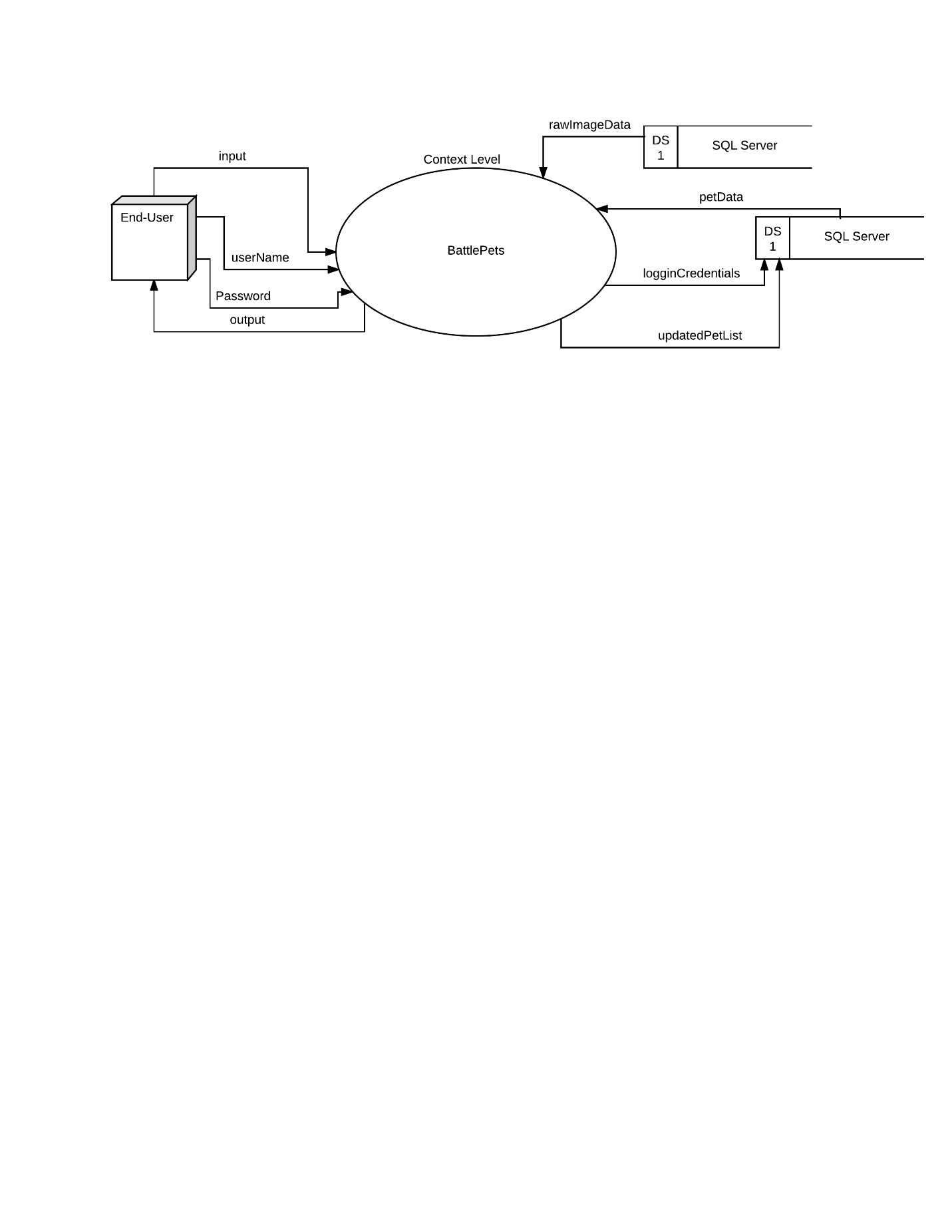
Standard Round:

Timestamp | Name | Ability/Switched/Attempted Capture | AttackHit/CaptureSuccessful

Finish:

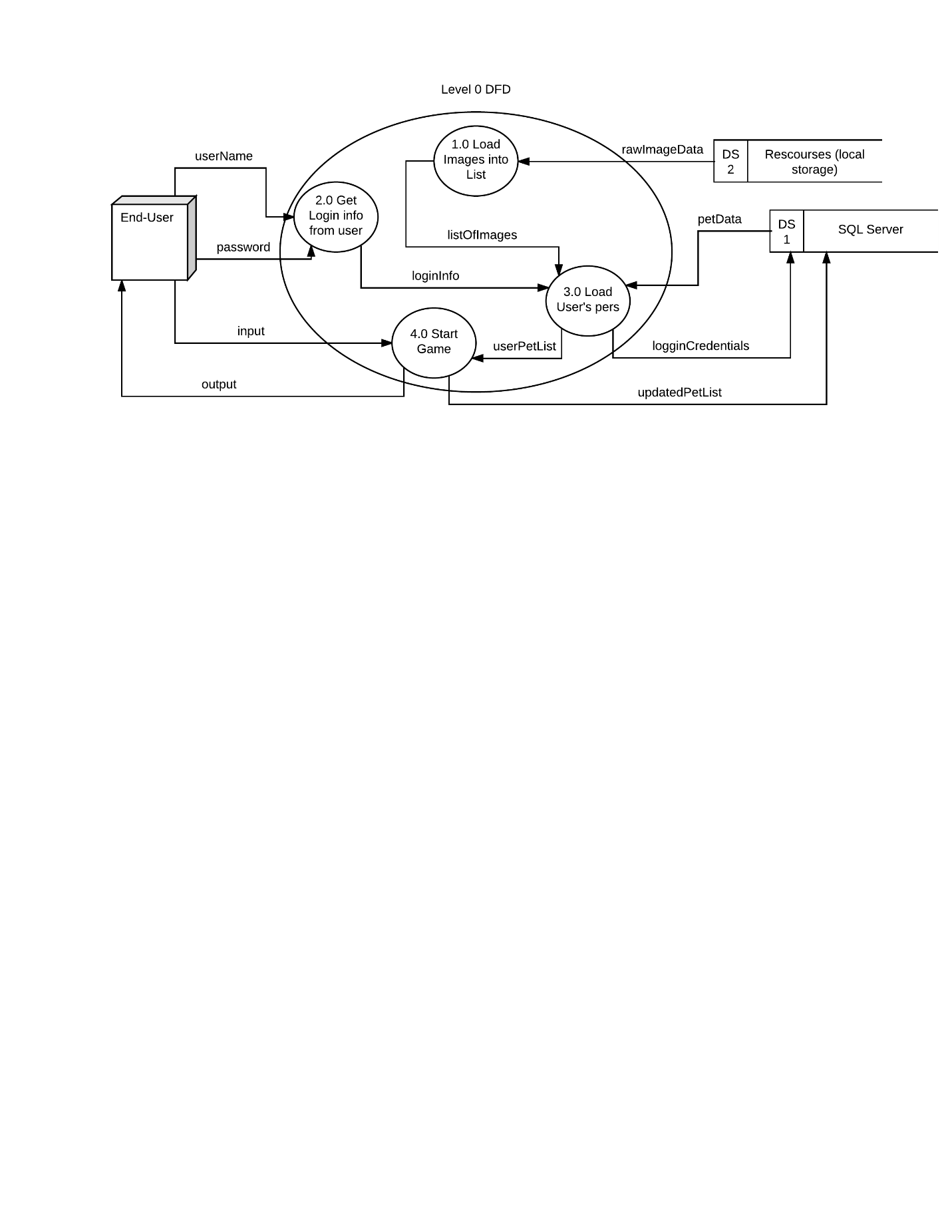
Timestamp | Name Fainted | Name Sent Out/ Battle Over

# Context Diagrams

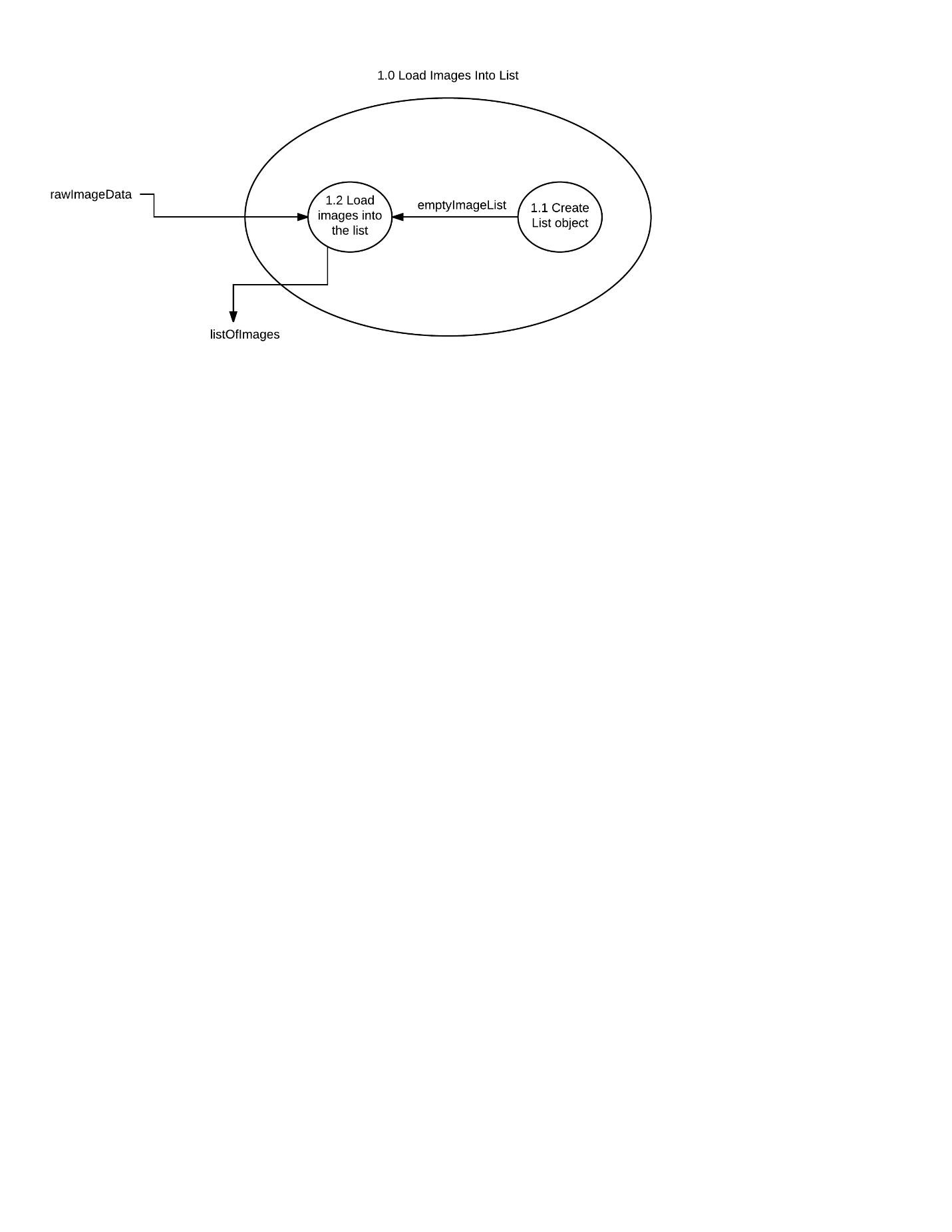


# Data Flows

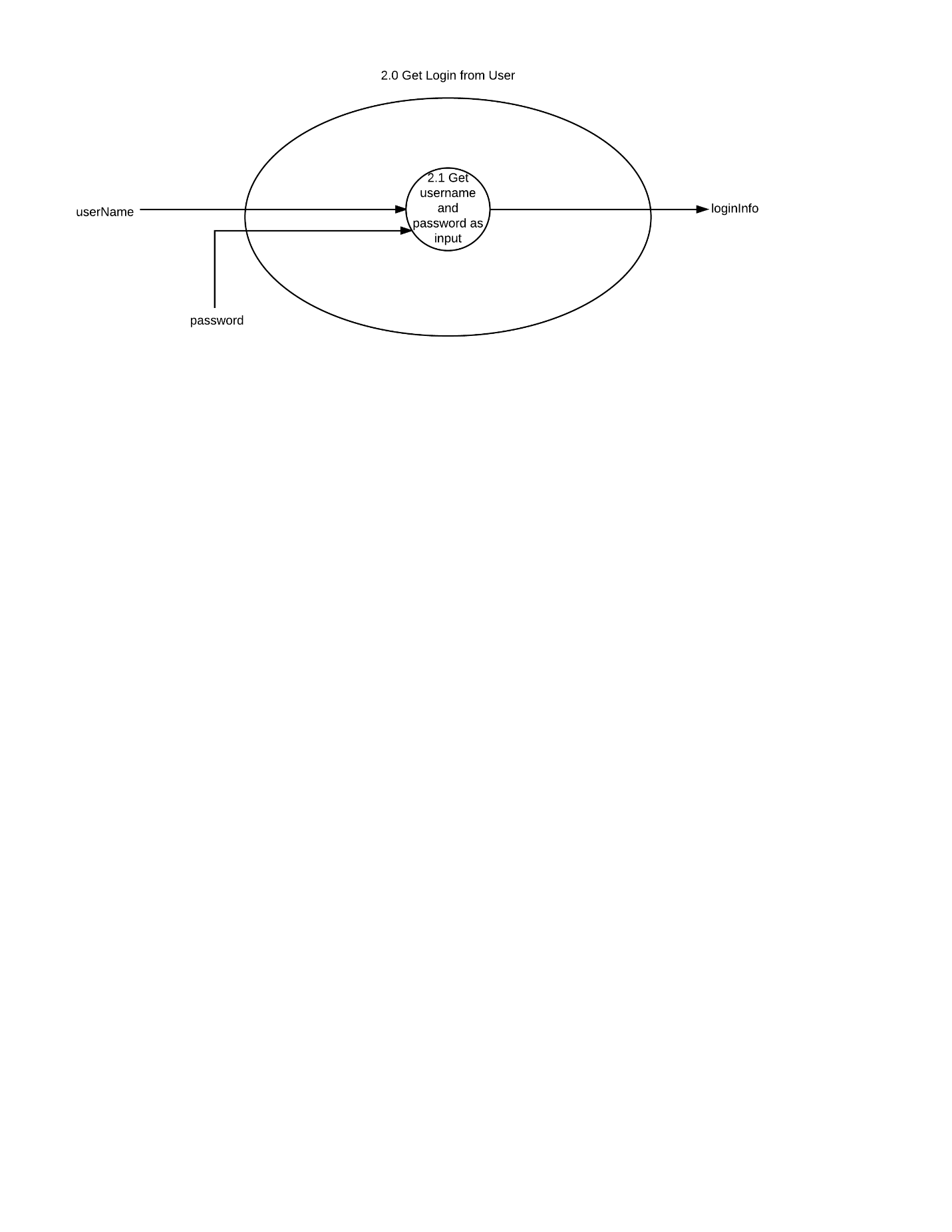
### Level 0



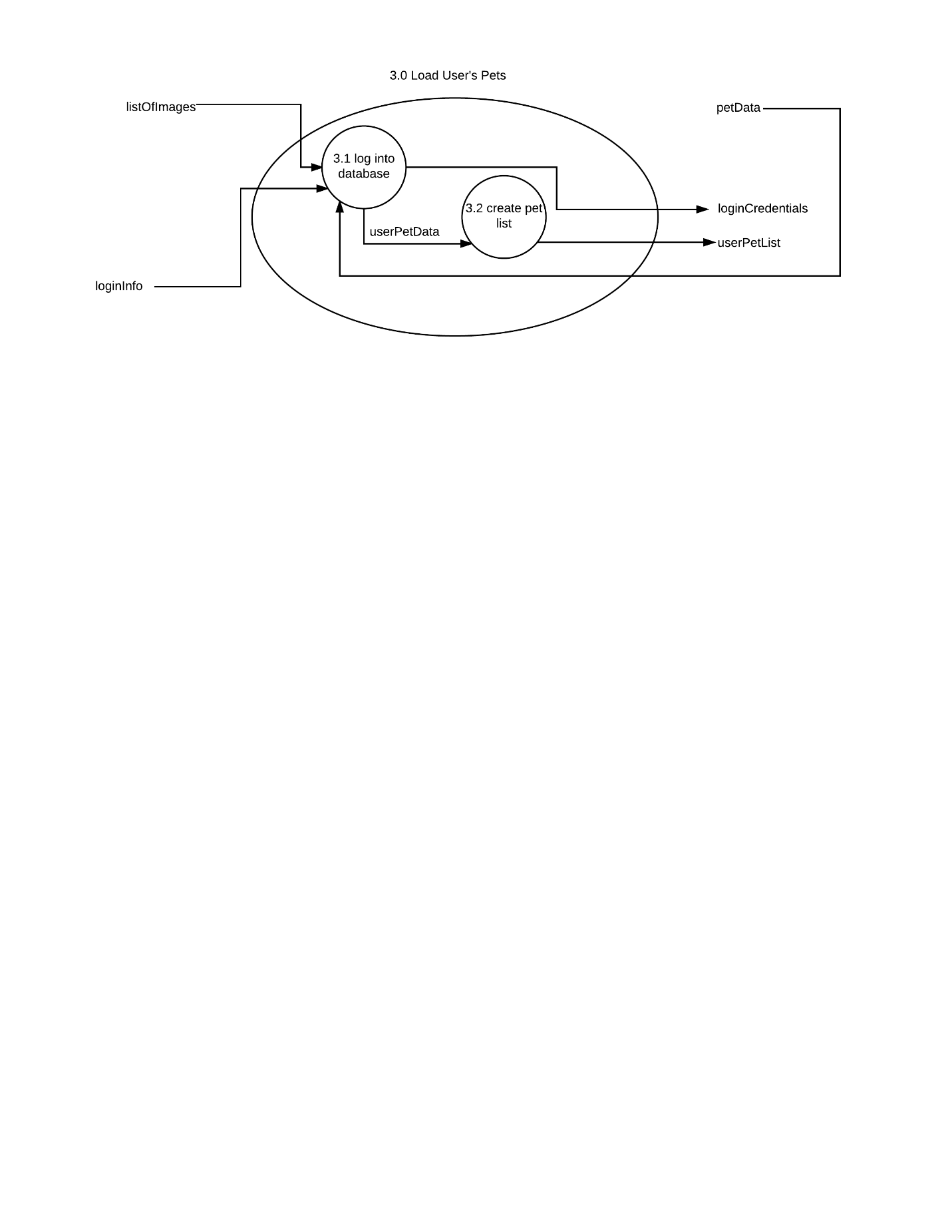
### 1.0



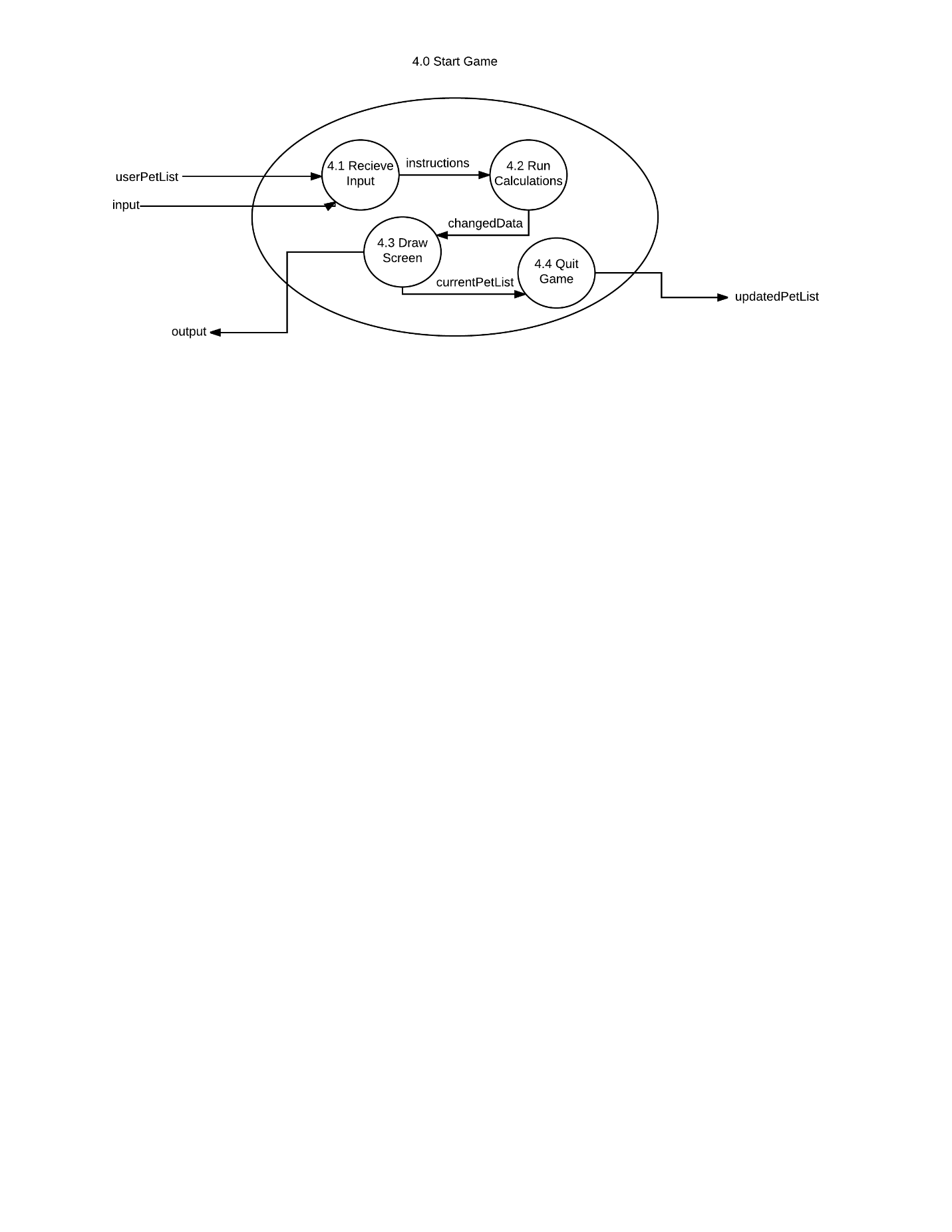
### 2.0



### 3.0



### 4.0



### Dictionary

#### Level 1

|  |  |
| --- | --- |
| Name | data type |
| rawImageData | Binary Image Data |
| emptyImageList | List<Image> |
| listOfImages | List<Image> |
|  |  |

#### Level 2

|  |  |
| --- | --- |
| Name | data type |
| userName | String |
| password | String |
| loginInfo | String[] |
|  |  |

#### Level 3

|  |  |
| --- | --- |
| Name | data type |
| ListOfImages | List<Image> |
| loginInfo | String[] |
| petData | Pet |
| userPetData | Pet |
| loginCredentials | String[] |
| userPetList | List<Pet> |

#### Level 4

|  |  |
| --- | --- |
| Name | data type |
| userPetList | List<Pet> |
| input | KeyCode |
| instructions | N/A |
| changedData | N/A |
| output | Screen render |
| currentPetList | List<Pet> |
| updatedPetList | List<Pet> |

## Hierarchy

GameObject

Player

Pet

Ability

Main Form

Starting Pet Form

New Game Form

Report Forms

Report Forms

Report Forms

If new game

## Blueprints

### GameObject

* String Name
* Point Location
* Image Image

### Pet

* Int ID
* String Name
* String Type
* String Rarity
* Int baseHP
* Int hpScale
* Int baseSpeed
* Int speedScale
* Int baseAtk
* Int attackScale
* Int currentXP
* Ability[] Abilities
* Int CurrentHP
* Ability selectedAbility
* Int Seed
* Int rosterSlot

### Ability

* String Name
* Int Damage
* Int Accuracy(out of 100)
* GetAbility()

### Player

* Int battlesWon
* Int battlesLost