Final Project Overview

Assignment instructions posted here

Mark Gryparis
PromineoTech Coding Bootcamp Homework
Cohort: 2023-04-26-be-pacific

GitHub Location:	**https://github.com/mgryparis/gameup**
Demo Video URL:	**https://youtu.be/Vn9XmUinTZs**
Project Participants:	Mark Gryparis (Solo Project)
Title:	GameUp (sung to the tune of MeetUp)

Executive Summary

Tabletop games are games played with multiple (physically present) players sitting around a table, often with a board and physical game pieces. They span the full range of scope and complexity from imagination-driven games like D&D & Warhammer 40K; to nerd-level board games like Scythe, Pandemic or Cathedral; to traditional games like Monopoly, chess or checkers.

GameUp is a MeetUp-style web application intended to enable tabletop gamers to:
- Connect with other tabletop gaming enthusiasts
- Organize, plan and execute gaming events, large and small

All online, in a self-serve/self-organizing way. This project will build a proof-of-concept back-end application to demo the core functionality, in a way that could be built upon with additional features and a web front-end to make it a real, deployable internet site/app.

GameUp Ontology:

GameUp's ontology consists of 4 core concepts (things):

Gamers: The humans who connect and self organize to plan and execute gaming events

Games: The types of games that Gamers are interested in playing (D&D, Pandemic, Monopoly, Chess,etc.)

Events: Scheduled Events in designated physical locations where Gamers meet to play Games

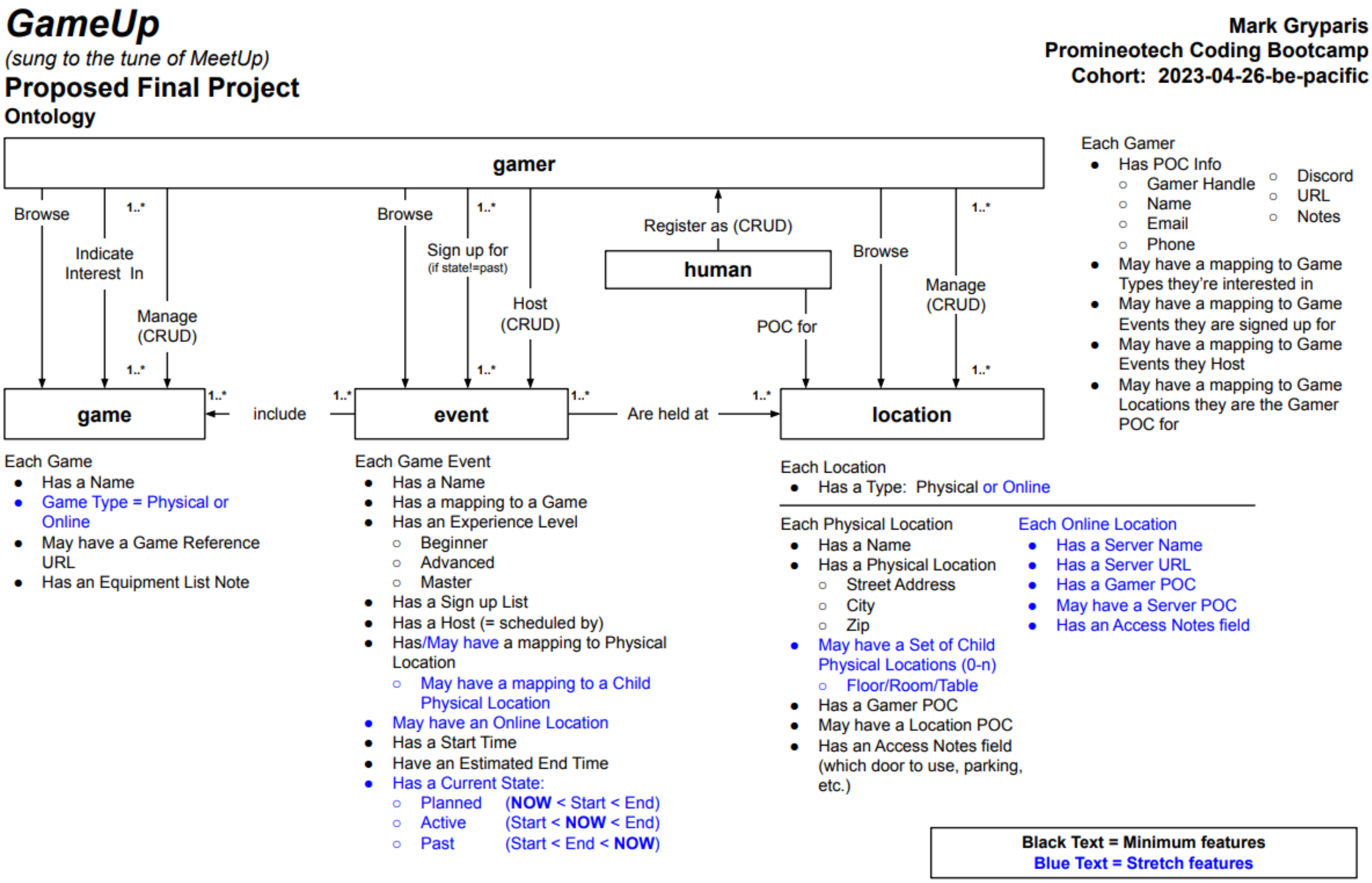
Locations: Physical Locations where Events can be held

The required behaviors/relationships/interactions (Use Cases) between these are:

1. Humans can register themselves as Gamers
    a. Admins are a subclass of Gamers with admin privileges
    b. Gamers can CRUD their own records, Admins can CRUD any Gamer
2. Gamers can register (create) Games (game types, e.g., chess)
    a. Gamers can update/delete Games they created, Admins can update/delete any Games
3. Gamers can review (read) the registered Games and indicate those they're interested in (CRUD a relationship to)
4. Gamers can register (create) Locations
    a. Gamers can update/delete Locations they created, Admins can update/delete any Location
5. Gamers can host (arrange/schedule) (create) Events of a selected Game at a Location
    a. Gamers can update/delete Events they created, Admins can update/delete any Event

6. Gamers can review the scheduled Events and sign up for those they're interested in attending (CRUD a relationship to)
7. Gamers, Games, and Locations exist and persist independently of each other and of Events
8. Events require a Host (Gamer), one or more Games and one or more Locations to exist
    a. If a, Event's Locations is reduced to zero, the Event is cancelled (deleted), which can happen when:
        i. A venue cancels (Event-to-Location mapping is deleted)
        ii. A venue is no longer available (Location is deleted)
    b. If a Event's Games is reduced to zero, the Event is cancelled (deleted), which can happen when
        i. A Game is no longer available at an event (Location-to-Game mapping is deleted)
        ii. A Game is no longer supported by GameUp (Game is deleted)
    c. If a Gamer is deleted, any Events hosted/scheduled by that Gamer are cancelled (deleted)
    d. If an Event loses its Host, the Event is cancelled (deleted), which can happen when:
        i. A Gamer is no longer available to Host an event (Gamer-to-Location mapping is deleted)
        ii. A Gamer leaves GameUp (Gamer is deleted)

Ontology Diagram



GameUp

(sung to the tune of MeetUp)

Proposed Final Project

Ontology

Mark Gryparis
Promineotech Coding Bootcamp
Cohort: 2023-04-26-be-pacific

Each Gamer

- Has POC Info
    - Gamer Handle
    - Name
    - Email
    - Phone
    - Discord
    - URL
    - Notes
- May have a mapping to Game Types they're interested in
- May have a mapping to Game Events they are signed up for
- May have a mapping to Game Events they Host
- May have a mapping to Game Locations they are the Gamer POC for

Each Game

- Has a Name
- Game Type = Physical or Online
- May have a Game Reference URL
- Has an Equipment List Note

Each Game Event

- Has a Name
- Has a mapping to a Game
- Has an Experience Level
    - Beginner
    - Advanced
    - Master
- Has a Sign up List
- Has a Host (= scheduled by)
- Has/May have a mapping to Physical Location
    - May have a mapping to a Child Physical Location
- May have an Online Location
- Has a Start Time
- Have an Estimated End Time
- Has a Current State:
    - Planned (**NOW** < Start < End)
    - Active (Start < **NOW** < End)
    - Past (Start < End < **NOW**)

Each Location

- Has a Type: Physical or Online

Each Physical Location

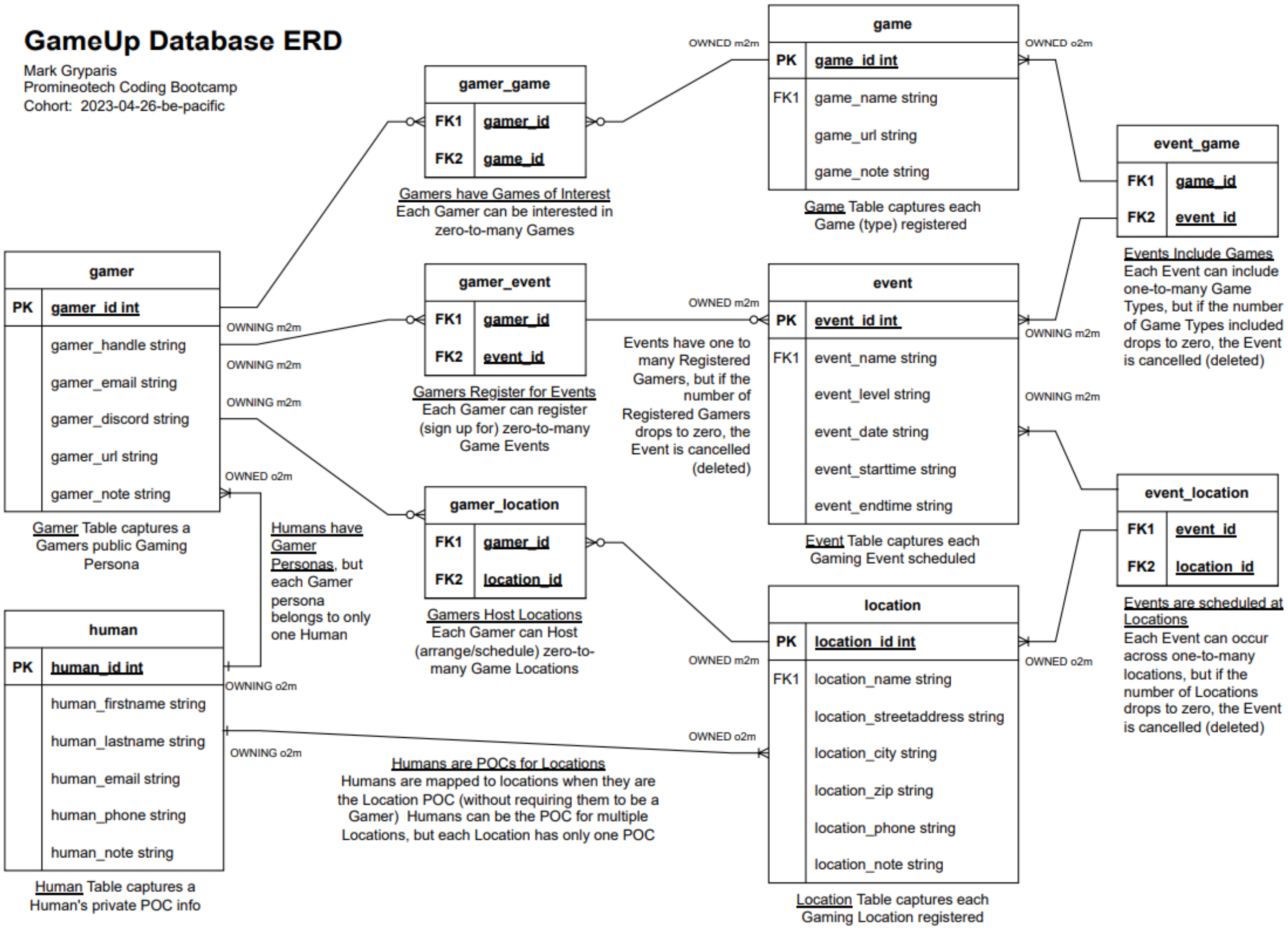
- Has a Name
- Has a Physical Location
    - Street Address
    - City
    - Zip
- May have a Set of Child Physical Locations (0-n)
    - Floor/Room/Table
- Has a Gamer POC
- May have a Location POC
- Has an Access Notes field (which door to use, parking, etc.)

Each Online Location

- Has a Server Name
- Has a Server URL
- Has a Gamer POC
- May have a Server POC
- Has an Access Notes field

Black Text = Minimum features
Blue Text = Stretch features

Database Design



GameUp Database ERD

Mark Gryparis
Promineotech Coding Bootcamp
Cohort: 2023-04-26-be-pacific

Gamers have Games of Interest
Each Gamer can be interested in zero-to-many Games

Game Table captures each Game (type) registered

Events Include Games
Each Event can include one-to-many Game Types, but if the number of Game Types included drops to zero, the Event is cancelled (deleted)

Gamer Table captures a Gamers public Gaming Persona

Gamers Register for Events
Each Gamer can register (sign up for) zero-to-many Game Events

Events have one to many Registered Gamers, but if the number of Registered Gamers drops to zero, the Event is cancelled (deleted)

Event Table captures each Gaming Event scheduled

Humans have Gamer Personas, but each Gamer persona belongs to only one Human

Gamers Host Locations
Each Gamer can Host (arrange/schedule) zero-to-many Game Locations

Events are scheduled at Locations
Each Event can occur across one-to-many locations, but if the number of Locations drops to zero, the Event is cancelled (deleted)

Humans are POCs for Locations
Humans are mapped to locations when they are the Location POC (without requiring them to be a Gamer) Humans can be the POC for multiple Locations, but each Location has only one POC

Human Table captures a Human's private POC info

Location Table captures each Gaming Location registered

Planned API Endpoints (not all endpoints Coded ;)

Base URL: **http://localhost:8080/gameup**

Endpoints:

Entity	***URL***	***Actions available at that endpoint***
Event	/event	**GET** all Events **POST** new Event
	/event/{id}	**GET** selected Event with its relationships **PUT** selected Event (update attributes) **DELETE** selected Event and its relationships
	/event/addGame	**POST** Event-to-Game relationship (“includes”)
	/event/addLocation	**POST** Event-to-Location relationship (“scheduled at”)
	/event/removeGame	**DELETE** Event-to-Game relationship (“includes”)
	/event/removeLocation	**DELETE** Event-to-Location relationship (“scheduled at”)
Game	/game	**GET** all Games **POST** new Game
	/game/{id}	**GET** selected Game with its relationships **PUT** selected Game (update attributes) **DELETE** selected Game and its relationships
Gamer	/gamer	**GET** all Gamers **POST** new Gamer
	/gamer/{id}	**GET** selected Gamer with its relationships **PUT** selected Gamer (update attributes) **DELETE** selected Gamer and its relationships
	/gamer/addGame	**POST** Gamer-to-Game relationship (“interested in”)
	/gamer/addEvent	**POST** Gamer-to-Event relationship (“registered for”)
	/gamer/addLocation	**POST** Gamer-to-Location relationship (“host for”)
	/gamer/removeGame	**DELETE** Gamer-to-Game relationship (“interested in”)
	/gamer/removeEvent	**DELETE** Gamer-to-Event relationship (“registered for”)
	/gamer/removeLocation	**DELETE** Gamer-to-Location relationship (“host for”)
Location	/location	**GET** all Locations **POST** new Location
	/location/{id}	**GET** selected Location with its relationships **PUT** selected Location (update attributes) **DELETE** selected Location and its relationships

Entity	***URL***	***Actions available at that endpoint***
Human	/human	**GET** all Humans **POST** new Human
	/human/{id}	**GET** selected Human with its relationships **PUT** selected Human (update attributes) **DELETE** selected Human and its relationships
	/human/addLocation	**PUT** Human-to-Location relationship (“POC for”)
	/human/addGamer	**PUT** Human-to-Gamer relationship (“identity of”)
	/human/removeLocation	**DELETE** Human-to-Location relationship (“POC for”)
	/human/removeGamer	**DELETE** Human-to-Gamer relationship (“identity of”)