Game Center Management System Documentation

1. Project Purpose and Functionality

The Game Center Management System is designed to assist users in locating game centers and arcade cabinets effectively. The system supports:

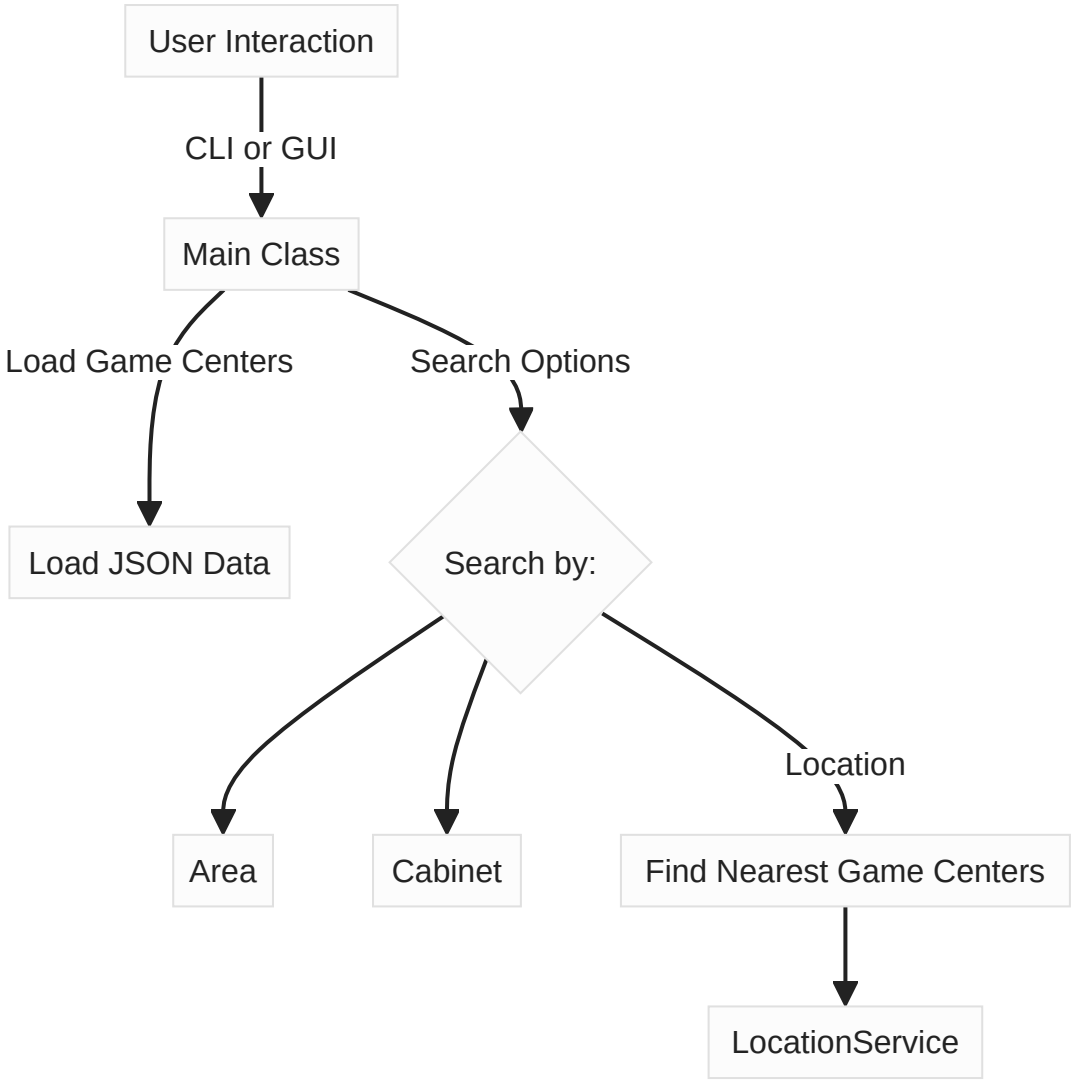
- Listing game centers and their details.
- Searching for specific arcade cabinets across various game centers.
- Managing arcade cabinet inventory within game centers.
- Viewing game center locations on a map and calculating distances.
- Graphical User Interface (GUI) for enhanced user interaction.

Key Functionalities

1. **Game Center Management:**
   - Use web scraper to get more game centers' data from the website
   - Search for game centers by the name of an arcade cabinet.
   - Display details of available game centers, including location, map URL, and distance from the user.
2. **Cabinet Management:**
   - Add new arcade cabinets with details such as name, description, manufacturer, genre, status, and price per play.
   - Check if a cabinet is operational.
   - View and update cabinet details.
3. **User Location Integration:**
   - Use Google’s Geolocation API to fetch the user’s location.
   - Calculate distances between the user’s location and game centers.
   - Display the nearest game centers.
4. **Graphical User Interface (GUI):**
   - JavaFX-based GUI for managing game centers and cabinets.
   - Interactive screens for searching by area or cabinet, viewing details, and locating nearest game centers.
5. **Future Features:**
   - **User Feedback:** Allow users to leave comments and verify the status of game centers and cabinets.

---

2. Explanation of Classes and Methods



Cabinet Class

The `Cabinet` class represents an arcade cabinet and contains the following fields and methods:

Fields:

- `name`: The name of the arcade cabinet.
- `description`: A brief description of the cabinet.
- `manufacturer`: The manufacturer of the cabinet.
- `genre`: The genre of games supported by the cabinet.
- `status`: Indicates whether the cabinet is operational (e.g., "Available").
- `pricePerPlay`: The cost to play the game (in yen).

Methods:

- `getName()`, `getDescription()`, `getManufacturer()`, `getGenre()`, `getStatus()`, `getPricePerPlay()`: Getter methods for retrieving cabinet details.

- `setName(String name)`, `setDescription(String description)`, `setManufacturer(String manufacturer)`, `setGenre(String genre)`, `setStatus(String status)`, `setPricePerPlay(int pricePerPlay)`: Setter methods for updating cabinet details.
- `isOperational()`: Checks if the cabinet is operational by evaluating its status.
- `toString()`: Returns a formatted string representation of the cabinet details.

GameCenter Class

The `GameCenter` class manages a collection of arcade cabinets within a specific location. It contains:

Fields:

- `name`: Name of the game center.
- `address`: Address of the game center.
- `area`: Area or region of the game center.
- `storeId`, `regionId`: Identifiers for the game center.
- `latitude`, `longitude`: Geographical coordinates.
- `distance`: Distance from the user's location.
- `mapUrl`: Google Maps URL for the location.
- `cabinets`: A list of arcade cabinets available in the game center.

Methods:

- `addCabinet(Cabinet cabinet, int quantity)`: Adds a specified quantity of a cabinet to the game center.
- `hasCabinet(String cabinetName)`: Checks if a specific cabinet exists in the game center.
- `printCabinets()`: Displays all cabinets in the game center.
- `toString()`: Returns a formatted string representation of the game center details.

Main Class

The `Main` class provides a Command-Line Interface (CLI) for user interaction. Key methods include:

- `loadGameCentersFromJson(String directoryPath)`: Loads game center data from JSON files.
- `searchGameCenterByArea(List<GameCenter> gameCenters, Scanner scanner)`: Allows users to search for game centers by area.
- `searchGameCenterByCabinet(List<GameCenter> gameCenters, Scanner scanner)`: Allows users to search for game centers by cabinet name.

- `findNearestGameCenters(List<GameCenter> gameCenters, double[] userLocation, Scanner scanner)`: Displays the nearest game centers based on the user's location.

LocationService Class

The `LocationService` class provides geolocation and distance calculation functionalities.

Methods:

- `getUserLocation()`: Fetches the user's current location using Google's Geolocation API.
- `calculateDistance(double lat1, double lon1, double lat2, double lon2)`: Calculates the distance between two geographical points.
- `getStaticMapUrl(double latitude, double longitude)`: Generates a Google Maps static map URL for the given coordinates.

MainGui Class

The `MainGui` class initializes and launches the JavaFX-based GUI.

- `start(Stage primaryStage)`: Loads the FXML file and sets up the main application window.

Controller Class

The `Controller` class manages the GUI logic and interaction.

Methods:

- `initialize()`: Loads game center data from JSON files.
- `handleSearchByCabinet()`: Handles the search functionality by cabinet name.
- `handleSearchByArea()`: Handles the search functionality by area.
- `handleFindNearestGameCenters()`: Displays the nearest game centers to the user's location.
- `showGameCenterDetails(GameCenter center)`: Displays detailed information about a selected game center.

---

3. JSON Data Integration

The system integrates external data through JSON files for managing game centers and cabinet details. An example JSON file for cabinets is provided:

Sample JSON File:

```
[
    {
        "name": "THE MOVE(ALIBABA)",
        "address": "73 Street, Between 105 x 107, Chan Mya Thar Si
Township",
        "game": "CHUNITHM International Version",
        "area": "Myanmar",
        "game_id": "104",
        "region_id": "1013",
        "store_id": "19187",
        "latitude": "21.9427482",
        "longitude": "96.0935648",
        "map_url": "https://maps.google.com/maps?q=THE+MOVE(ALIBABA)"
    },
    {
        "name": "ALIBABA 7",
        "address": "4TH FLOOR, JUNCTION CITY, PABEDAN TOWNSHIP,YANGON",
        "game": "CHUNITHM International Version",
        "area": "Myanmar",
        "game_id": "104",
        "region_id": "1013",
        "store_id": "19694",
        "latitude": "16.7790825",
        "longitude": "96.1543205",
        "map_url": "https://maps.google.com/maps?q=ALIBABA+7"
    }
]
```


This data can be parsed and used to populate the system with real-world game center information.

---

4. Future Plans

- **Enhanced GUI:** Add more interactive features to the JavaFX interface.
- **Feedback System:** Allow users to review game centers and report the operational status of cabinets.