IEE 305 – Term Project Proposal

Park Operations Management System (POMS)

Course Name: Information System Engineering

Assignment Due Date: 25/11/2025

Group Information

Name	Email	Responsibility
Hamad Alnuaimi	hsalnua1@asu.edu	ER model & relational schema design
		Backend FastAPI development, SQL queries
		Frontend UI and integration with backend
		Data acquisition, transformation, documentation

Problem Statement

National Park Service (NPS) operations involve complex resource management across multiple parks, visitor centers, and campgrounds. Managers must allocate staff, schedule maintenance, and ensure facilities meet visitor demand while balancing budget and safety constraints. Currently, data on park operations, visitor numbers, and facilities is scattered and not optimized for decision-making. Due to this, administrators cannot respond to challenges encountered throughout the operational process. The inconsistent data sources cause slow coordination between departments, prompting managers to rely on information from manual spreadsheets. Increased dependence on manual spreadsheets has exposed them to inefficiency risks, missed opportunities, and underutilized resources. Based on this, implementing a data-driven model will improve planning accuracy, efficiency, and operational transparency.

IE Problem to Solve

Park operations managers need to answer questions such as:

- Which parks or visitor centers are under or over capacity on a given date?
- How should maintenance or staffing be prioritized across seasonal variations?
- Which activities attract the most visitors, and which facilities require upgrades?

Engineering Value

The National Park Service API provides structured data on park designations, facilities, and visitor activities. Using this API allows for:

- Evidence-based decision-making; the API provides well-organized data and therefore managers should use factual information when making decision instead of fragmented reports.
- Capacity planning using real-world visitor data; the API incorporate different metrics like activity participation, defective usage and visitation numbers. The engineers can evaluate expected demand, make future prediction and approximate peak loads by analyzing API patterns.
- Prioritization of maintenance or staffing based on activity and usage trends; API help determine the frequency of activities, trails and facilities. Therefore, planners the point to expect wear-and -tear and proactively allocate management crews.

Target Users

The system is used by NPS operational managers, visitor experience coordinators and facility planners. The facility planners help in maintaining needs and access usage, coordinators examine trends that improves experience quality and managers help monitor staffing and capacity.

Entity-Relationship Model

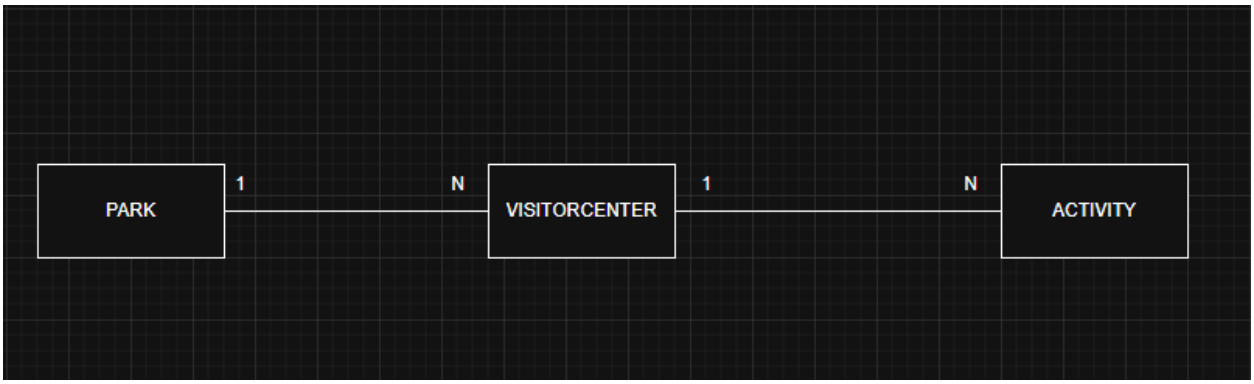
Entities:

Park – Contains park-level information

VisitorCenter – Facilities within each park

Activity – Recreational or educational activities offered

ER Diagram



Attributes:

Park: ParkID (PK), Name, State, Designation, Description, EntranceFee

VisitorCenter: CenterID (PK), ParkID (FK), Name, OperatingHours, ContactPhone, ContactEmail

Activity: ActivityID (PK), CenterID (FK), Name, Category, Availability

Relationships:

1. One Park can have many Visitor Centers (1:N): This relationship infers that a certain park contains several centers. However, every visitor has only one park.
2. One Visitor Center can host many Activities (1:N): This relationship denotes that a single center can provide several activities like wildlife tours, educational sections of guided hikes, but every activity is only held for one visitor center.

The above relationships are presented with (1:N), indicating that one unit can relate to one unit from several units on right. There can be several units from left side but each will relate to one situated on the left side. These relationships ensure efficient reporting, decision-making and querying based on resource allocation and park operations.

Relational Data Model

Table: **Park**

Column	Data Type	Constraints
ParkID	INTEGER	PRIMARY KEY
Name	TEXT	NOT NULL
State	TEXT	NOT NULL
Designation	TEXT	NOT NULL
Description	TEXT	
EntranceFee	TEXT	

Table: **VisitorCenter**

Column	Data Type	Constraints
CenterID	INTEGER	PRIMARY KEY
ParkID	INTEGER	NOT NULL, FOREIGN KEY REFERENCES Park(ParkID)
Name	TEXT	NOT NULL
OperatingHours	TEXT	
ContactPhone	TEXT	
ContactEmail	TEXT	

Table: **Activity**

Column	Data Type	Constraints
ActivityID	INTEGER	PRIMARY KEY
CenterID	INTEGER	NOT NULL, FOREIGN KEY REFERENCES VisitorCenter(CenterID)
Name	TEXT	NOT NULL
Category	TEXT	NOT NULL
Availability	TEXT	

Normalization Justification:

1NF: All attributes are atomic; this means that there is no group or array repetition. Each activity has its separate document with single availability and category.

2NF: No partial dependencies; all non-key attributes depend on the full primary key. For instance, in a given visitor center, the hours of operations are based on the ID of the center but not just the ID of the park.

3NF: No transitive dependencies; all attributes depend only on the primary key. For example, several parks feature like designation and state will depend on the park ID. On the other side the activity depends on the activity ID.

The foreign keys enforce table relationship and help to uphold integrity across database. Every center must be connected to a given park for park ID to have valid park record. The activity of visitor is linked to certain center through center ID which preserves hierarchy of centers, activities and parks. Constraints like NOT NULL improves data integrity and confirms that significant fields are not left empty. A unique constraint there will prevent duplicate entries so that a park appears once in the database. Through this, the park will ensure reliability and accuracy of dataset and simplify reporting and querying processes. Managers can retrieve aggregated statistics, provide reports and evaluate high-demand activities for effective decision making.

Sample Scraped Data

Park Table:

ParkID	Name	State	Designation	Description	EntranceFee
1	Abraham Lincoln Birthplace	KY	National Historical Park	Honors early life of Lincoln	None
2	Acadia	ME	National Park	Protects Atlantic coastline	$35 vehicle
3	Adams NHP	MA	National Historical Park	Preserves Adams family homes	$15/person

VisitorCenter Table:

CenterID	ParkID	Name	OperatingHours	ContactPhone	ContactEmail
1	1	Birthplace Unit	9 AM–5 PM	270-358-3137	ABLI_Administration@nps.gov
2	1	Boyhood Unit	Sunrise–Sunset	270-358-3137	ABLI_Administration@nps.gov
3	2	Hulls Cove Visitor Center	8 AM–5 PM	207-288-3338	acadia_information@nps.gov

Activity Table:

ActivityID	CenterID	Name	Category	Availability
1	1	Guided Tours	Tour	Year-round
2	2	Junior Ranger Program	Education	Year-round
3	3	Hiking	Recreation	Year-round

API RESPONSE:

Park

Curl

```
curl -X GET "https://developer.nps.gov/api/v1/parks?limit=3&api_key=cIDFJCbbnZPJQkcPAnOJ1Q9DCT4dT6PkJeB8We4b" -H "accept: application/json"
```

Server response

Code	Details
200	Response body


```
{
  "total": "474",
  "limit": "3",
  "start": "0",
  "data": [
    {
      "id": "77E0D7F0-1942-494A-ACE2-9004D2BDC59E",
      "url": "https://www.nps.gov/abli/index.htm",
      "fullName": "Abraham Lincoln Birthplace National Historical Park",
      "parkCode": "abli",
      "description": "For over a century people from around the world have come to rural Central Kentucky to honor the humble beginnings of our 16th president, Abraham Lincoln. His early life on Kentucky's frontier shaped his character and prepared him to lead the nation through Civil War. Visit our country's first memorial to Lincoln, built with donations from young and old, and the site of his childhood home.",
      "latitude": "37.5858662",
      "longitude": "-85.67330523",
```


Activities

Curl

```
curl -X GET "https://developer.nps.gov/api/v1/activities?id=1&api_key=cIDFJCbbnZPJQkcPAnOJ1Q9DCT4dT6PkJeB8We4b" -H "accept: application/json"
```

Server response

Code	Details
200	Response body


```
{
  "total": "40",
  "limit": "50",
  "start": "0",
  "data": [
    {
      "id": "09DF0950-D319-4557-A57E-04CD2F63FF42",
      "name": "Arts and Culture"
    },
    {
      "id": "13A57703-BB1A-41A2-94B8-53B692EB7238",
      "name": "Astronomy"
    },
```


VisitorCenter

Curl

```
curl -X GET "https://developer.nps.gov/api/v1/visitorcenters?limit=1&api_key=cIDFJCbbnZPJQkcPAnOJ1Q9DCT4dT6PkJeB8We4b" -H "accept: application/json"
```

Server response

Code	Details
200	Response body


```
{
  "total": "712",
  "limit": "1",
  "start": "0",
  "data": [
    {
      "id": "6DC6F87E-5D33-469F-8C9C-6D821186633A",
      "url": "",
      "name": "A.G. Gaston Motel",
      "parkCode": "bicr",
      "description": "The A.G. Gaston Motel served as the headquarters for the Birmingham campaign. In April through May of1963 civil rights leaders, including Reverend Dr. Martin Luther King Jr., took up residence at the motel where they strategized and made critical decisions about the non-violent campaign that targeted Birmingham's segregation laws and practices. Several key events of the campaign publicly unfolded at the property.",
      "latitude": "33.51516726831536",
      "longitude": "-86.81467831134796",
      "latLong": "{lat:33.51516726831536, lng:-86.81467831134796}",
```

SQL Query Specifications

Q1.	List all parks in a specific state	SELECT with WHERE
Q2.	Count visitor centers per park	JOIN + COUNT
Q3.	List activities offered at a specific park	2-table JOIN + WHERE
Q4.	Count activities per category per park	JOIN + GROUP BY
Q5.	Find parks with more than 3 activities	HAVING + GROUP BY
Q6.	List top 5 parks by number of visitor centers	ORDER BY + LIMIT
Q7.	Find all activities not available year-round	Subquery + WHERE
Q8.	List visitor centers and their park names	JOIN + SELECT
Q9.	Calculate average number of activities per visitor center	Aggregation + JOIN
Q10.	Parameterized query for activity search by name	Parameterized WHERE clause

API Integration Plan

API: National Park Service API

Endpoints: /parks, /visitorcenters, /activities

Data Fetch Strategy:

This strategy performs the initial one-time fetch, populating SQLite database with organized data concerning visitor centers, activities and parks. Based on this, all entities will

have a representative data for analysis. In some scenarios, we can schedule periodic updates to capture updated visions hours, change available activity and new park, which help to maintain accuracy. Error handling and data validation help to confirm that only consistent and clean data are inserted.

Mapping to Schema

parkCode -> ParkID, fullName -> Name, states -> State, designation -> Designation

VisitorCenter fields mapped to CenterID, Name, OperatingHours.

Activity mapped from activity lists, category, and availability

The mapping helps to transform API data into normalized tables, and manage relationships between parks and other activities for reporting and analytical queries. Maintaining such relationship is essential during decision making about activities at the park.

Technology Stack

Backend: Python 3.12.1, FastAPI, SQLModel indispensable for ORM interactions with SQLite.

Frontend: HTMLS, CSS and Javascript to develop interactive and responsive interface.

Data Validation: Pydantic which ensures data integrity between database and API

Version Control: Git/GitHub useful in change tracking and collaborative development.

Database: SQLite (3 tables, 3NF, indexed foreign keys) which is used in optimizing query performance.

References

Kumar, S. (2019). A Review on Client-Server based applications and research opportunity. *International Journal of Recent Scientific Research*, *10*(7), 33857-3386.

Ofoeda, J., Boateng, R., & Effah, J. (2019). Application programming interface (API) research: A review of the past to inform the future. *International Journal of Enterprise Information Systems (IJEIS)*, *15*(3), 76-95.

Melton, J. (2006). Database language sql. In *Handbook on Architectures of Information Systems* (pp. 103-128). Berlin, Heidelberg: Springer Berlin Heidelberg.