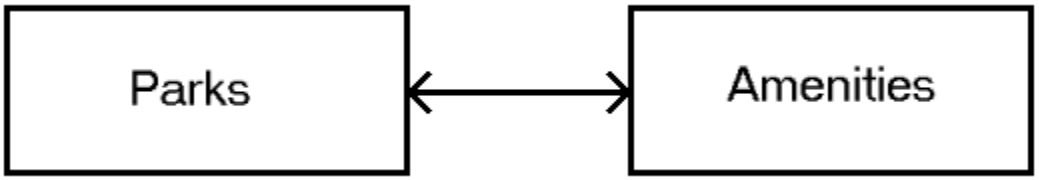
Creative Collaboration Project

The Creative Collaboration project is an extension on the Passion Project, collaborating your passion with one or more of your peers. It is a group project, however, you are marked based on your individual contribution.

Example: A creative Collab between Sam and Alex. Sam has a project on parks, Alex has a project on music. Together, they create a collaboration project for which songs pair well with which parks.

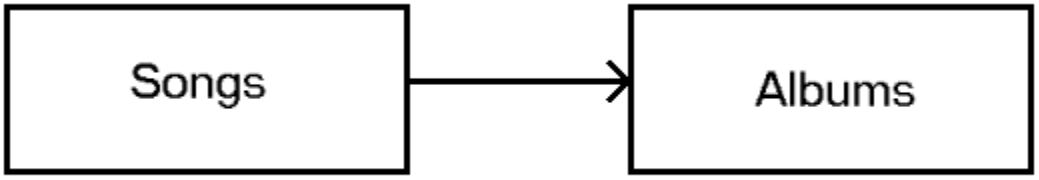
Sam's Passion Project: Parks

- Create, Read, Update, Delete Parks
- Create, Read, Update, Delete Amenities
- Related CRUD
    - ListParksForAmenity
    - ListAmenitiesForPark
    - AddAmenityToPark
    - DeleteAmenityFromPark



Alex's Passion Project: Songs

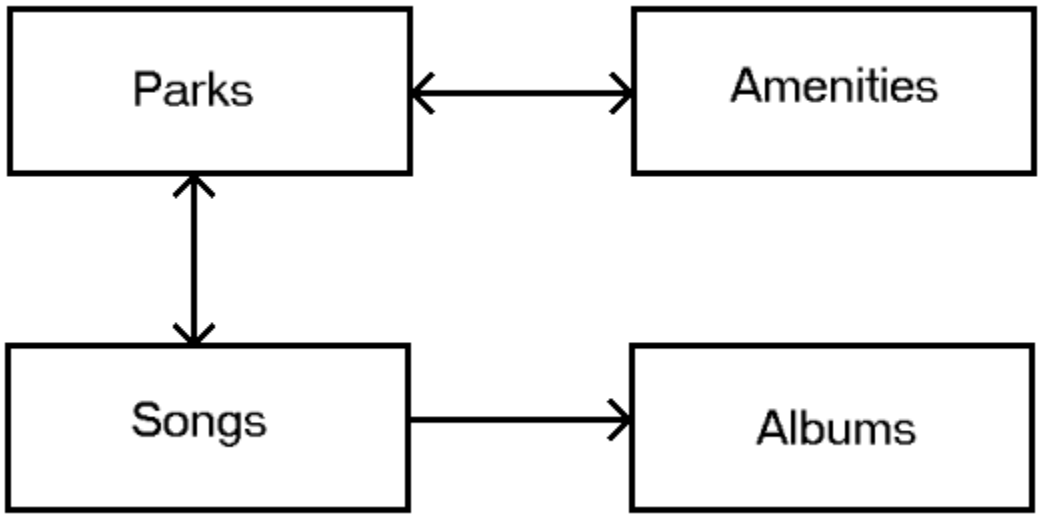
- Create, Read, Update, Delete Songs
- Create, Read, Update, Delete Albums
- Related CRUD
    - ListSongsForAlbum



Example (Cont'd)

Sam & Alex Creative Collab: Songs & Parks

- Create, Read, Update, Delete Parks
- Create, Read, Update, Delete Amenities
- Create, Read, Update, Delete Songs
- Create, Read, Update, Delete Albums
- Related CRUD
    - ListParksForAmenity
    - ListAmenitiesForPark
    - AddAmenityToPark
    - DeleteAmenityFromPark
    - ListSongsForAlbum
    - ListSongsForPark
    - ListParksForSong
    - AddSongToPark
    - RemoveSongFromPark



The collaboration would exist as a new project between Sam and Alex.

Team Meeting (⅓ of Project Total)

The meeting is where you will coordinate your collaboration idea, entity relationship diagram, wireframes, and overall plan.

Team	Team size: minimum 2, maximum 3.
ERD	● Unified 'Master' Entity Relationship Diagram which shows all of the team's entities and their connections (see Collab Example)
Wireframes	● Wireframes reflect Create, Read, Update, and Delete for your entities. ● Wireframes reflect how a user can Create, Read, Update, and Delete relationships between entities. ● An indication of which actions are meant for admin-only (logged in).
Schedule	● An established timeline with the following goals: ○ New Repository Created ○ Database Built (with new connections) ○ Passion Project MVP completed / loaded in ○ Creative Collab MVP ○ Documentation (API + .readme) ○ Admin Functionality ○ Extra Feature ● When the team will be able to meet
Troubleshooting Strategies	Plans on how to resolve issues in the following scenarios: ● Migration Issues ● Git Conflicts ● CRUD via WebAPI not working ● CRUD via Controllers/Views not working

Level 1 (0-25%)	Level 2 (25-50%)	Level 3 (50-75%)	Level 4 (75%-100%)
Does not participate in a project plan meeting with the instructor. Unable to start the MVP.	Not prepared for the meeting. Missing multiple required elements. Significant revisions required before starting the MVP.	Prepared for the meeting. Some revisions to the plan required. Almost ready to begin the MVP.	Extremely prepared for the meeting. Ready to begin the MVP immediately.

Minimum Viable Product (⅓ of Project Total)

Based on feedback from the team meeting, work towards a minimum viable product for full Create, Read, Update, and Delete functionality.

- Creating Code-First models with 1-M or M-M relationships according to your design.
- Build a WebAPI Controller with the following data access methods:
    - List entities in your database
    - Find an individual entity in the database by it's ID
    - Create a New entity in the database given POST input
    - Update an entity in the database given POST input and an ID
    - Delete an entity in the database given an ID
    - List associated records given an entity ID
    - Add a new association to a record
    - Delete an association from a record
- Build a Controller, Views, and ViewModels which render the following pages:
    - Listing all Entities
    - Showing an entity (and entities related to it)
    - Creating a new entity (and creating associations to other entities)
    - Updating an entity (and updating associations to other entities)
    - Deleting an entity

Level 1 (0-25%)	Level 2 (25-50%)	Level 3 (50-75%)	Level 4 (75%-100%)
Feedback from the project plan is not incorporated. Few elements of Create, Read, Update and Delete implemented.	Feedback from the project plan is partially incorporated. Code does not meet professional standards. Create, Read, Update and Delete is partially implemented. Significant revisions required before starting on extra features.	Feedback from the project plan is mostly incorporated. Create, Read, Update and Delete is implemented for the base entities. Code is close to professional standards. Almost ready to begin on extra features.	Feedback from the project plan is fully incorporated. Create, Read, Update, and Delete is fully implemented for the base entities as well as relationships. Code meets professional standards. Ready to begin on extra features.

Finale (⅓ of Project Total)

After building the Minimum Viable Product, use feedback provided to add extra dimensions to your work, including admin functionality, extra features, and a strong documentation effort.

Finale Requirements

- Your methods which Add, Update, and Delete entities in your database will only execute if the user is logged in as an administrator.
- An attempt at an additional feature* explored in class, or your own choice of extra feature
- Your work is well documented on the Project github .readme

*Examples of an additional feature

- Use of JavaScript to validate inputs
- Use of JavaScript and AJAX in conjunction with a CRUD element
- Customized HTML/CSS
- Image Uploading
- Pagination / Search for List View
- Styling & Adding dynamic content to the home page (Shared/_Layout.cshtml)
- Connecting to an external API with C# HttpClient
- Assisting fellow teammates with Debugging/Questions
- .. Any other way you'd like to improve your project, talk with me about it!

Level 1 (0-25%)	Level 2 (25-50%)	Level 3 (50-75%)	Level 4 (75%-100%)
MVP not completed/improved upon based on feedback. No additional feature or admin functionality. No project .readme. Little to no communication with group members and teacher.	Feedback from the MVP is partially incorporated. Limited documentation. No extra feature attempted. Limited communication with group members and teacher.	Feedback from the MVP is mostly incorporated. Documentation is included but could be improved. An extra feature is done. Satisfactory communication with group members and teacher.	Feedback from the MVP is fully incorporated. Clear and concise documentation. Extra features are explored in depth. Strong communication with group members and teacher.

Summary of Creative Collaboration Requirements:

Stage	Requirement	Technique
Meeting	The application is planned, in terms of how a user and admin would navigate the site. Includes realistic examples.	Wireframe
Meeting	Relational database is planned, including all entities and their associations with one another.	Entity Relational Diagram
Meeting	Estimated project timeline, deliverables, and available meeting times	Team Coordination
Meeting	Debugging strategies, troubleshooting techniques	Team Coordination
MVP	The content in the system is managed through a relational database	Entity Framework, Code First Migrations
MVP	The website can access and modify information from the database upon request. Access is well documented	WebAPIControllers, Language Integration Query (LINQ) Methods. Summary blocks
MVP	The website can render information from the database into web pages	Controllers, Views, ViewModels, Http Client
MVP	At least one connection (1-M) or (M-M) between your entity and your teammates'	Entity Framework, Code First Migrations, WebAPIControllers, Language Integrated Query (LINQ) Methods, team coordination
Finale	Information in the database can be manipulated by an administrator who is logged in	ApplicationUser, Authorize Attributes
Finale	Exploration of an additional feature on top of Create, Read, Update, and Delete, such as: ● Image / File Uploading ● CSS / Styling ● Pagination of records ● Theming ● External API ● .. or your own idea for an additional feature!	Refer to in class examples
Finale	Project is well documented on github	github .readme, markdown

Code Standards

After writing a working solution, always take a moment to re-evaluate your code based on these standards. Make sure to document your codebase, practice debugging strategies, choose readable variable names, factor code to reduce redundancy, etc.

	Level 1 (0-25%)	Level 2 (25-50%)	Level 3 (50-75%)	Level 4 (75-100%)
Code Standards Quantitative	Codebase does not reach the desired goal for MVP or Additions.	Codebase reaches the goal of the MVP and some additions. Major concerns with scalability, maintainability, extensibility, robustness, or efficiency.	Codebase reaches the goal of MVP and 1-2 additions. Some concerns with scalability, maintainability, extensibility, robustness, or efficiency.	Codebase reaches the goal of the MVP and 3-4 additions . No major concerns with scalability, maintainability, extensibility, robustness, or efficiency.
Code Standards Qualitative	Code and Project is not documented.	Code and project are documented, but there are significant readability concerns.	Code and project are documented. Readability could be improved.	Code and project are meticulously documented. Readability is a high priority to the project.
Code Standards Semantic	Project does not use conventions described in class examples. Codebase is completely different from the original plan.	Project heavily deviates from conventions described in class examples. Codebase differs from the original plan.	Project slightly deviates from conventions described in class examples. Codebase resembles the initial plan.	Project closely follows conventions described in class examples. Codebase is a close reflection of the initial plan.