Unmanned Surface Vessel (USV) Fleet Management System

Alexander Jones and Alexander Lane
Team 45

Feedback by the TAs and Peer Reviewer

Reviewer: Sy Soto

Looks like a well-scoped design, and the ERD matches the written description clearly. The use of intersection tables for both USV–Payloads and CrewMembers–Qualifications makes sense and shows a good understanding of M:M relationships from what I've read and undertstand. One thing to think about is whether CrewMembers should really be limited to a single “primary” USV via a 1:M relationship, since in real operations sailors might rotate or be temporarily assigned elsewhere. Solid work from my point of view.

Reviewer: Charles Gilbert

• The overview does a great job explaining the problem. The problem being that the Navy is tracking USV crews and payloads manually with multiple files, which can easily get inconsistent. This idea of building a centralized database to manage USV status, crew assignments, and payload configurations makes a lot of sense. One small suggestion would be to include an example question the system could answer, such as: “Which deployable USVs currently have a sonar payload installed and a qualified operator assigned?” (if that’s something this interface would support).

• The overview also includes useful specific facts, for example, that there are about 25 USVs, each capable of carrying multiple payloads, along with roughly 150 crew members and many mission packages. These details help show why a database is needed instead of spreadsheets.

• They describe more than four entities, so they definitely meet the assignment requirement. Their entities include: USVs, Payloads, USVPayloads (the junction table), CrewMembers, Qualifications, CrewMemberQualifications (another junction table), and Missions. I don’t see any major issues here, but I’d just caution them not to keep expanding the scope too much with additional tables.

• The database outline is very clear. For each entity, they explain what it represents, list attributes with data types, and include constraints like PK, FK, NOT NULL, and UNIQUE. I also liked that they explained design choices. For instance, using a “status” field in USVs to track readiness (deployable, maintenance, etc.). One minor thought is that CrewMembers → USVs is modeled as 1:M, but in real life a sailor might work on multiple vessels over time. That could eventually be modeled as M:M, but for this class, keeping it simple is totally reasonable.

• The relationships look correct overall. There are clear 1:M relationships and at least one M:M relationship (USVs ↔ Payloads, and CrewMembers ↔ Qualifications). The ERD matches the written outline and seems logically sound.

• Naming is mostly consistent. The entities are plural (USVs, Payloads, CrewMembers, Qualifications, Missions) and attributes are singular. A small stylistic suggestion would be to rename USVPayloads to USV_Payloads for readability, and maybe shorten CrewMemberQualifications to Crew_Qualifications, but these are just cosmetic changes.

• Overall, this is a strong, realistic, and well-structured project. The problem is clearly motivated, the entities make sense, the relationships are logical, and the ERD aligns nicely with the written outline. Great Job!

Originality: This review is my own analysis. I used AI (ChatGPT) only to check my writing for major grammatical or clarity issues with the prompt: "Please review my writing for grammatical issues that would cause clarity problems."

Reviewer: Jordan Smith

Hello Alexanders (Sorry),

This overview hurts my soul, only because it hits a little too close to home with my current job. I unfortunately am completely able to understand why this database would be needed. "*tracked by hand and usually there are many versions of the same file floating around, with minor or major differences*" these words cut deep…

I see plenty of specific facts: 25 Unmanned USV's, ~150 feet long, 150 crew, payload capacity, 150 mission packages. I think you both covered quite well the generic facts that would be needed/included. Maybe include the potential number of qualifications? Also, maybe considered using different numbers? It looks like 150 is someone's favorite number. I usually try to keep all my numbers fairly different (order of magnitude if possible), that way I can just eyeball real quick to see if things are making sense.

Plenty of entities are included in this proposal: USVs, Payloads, CrewMembers, Qualifications, and Missions. Plus a few intersection tables. All of these entities make sense and appear to have reasonable attributes with the correct datatypes. The corresponding relationship descriptions also look logical to me.

Relationships appear correct to me. They follow a logical pathway that is easy to follow along with. Numerous M:M relationships are present.

Everything looks consistent to me. Entity names are consistent along with attribute formats. No recommendations.

Great work! I have no major recommendations, only minor personal preferences.

Everything above is my own work/effort.

Reviewer: Mauricio Marin Gutierrez

Hello Alexander Jones and Alexander Lane,

Thank you for sharing your well-organized project draft. I have reviewed it and will provide my feedback by answering the required questions.

The overview clearly describes the problem to be solved. The US Navy has a growing fleet of unmanned vessels, and their crew assignments, mission packages, and vessel status are tracked manually on paper. This leads to different versions of files and inefficiencies. Your proposed website with a database backend aims to centralize this tracking.

The overview lists specific facts that illustrate the scope. You mention twenty-five USVs, around one hundred fifty crew members, and nearly one hundred fifty different mission packages. These facts show the project manages a significant amount of interconnected data, which strongly justifies the need for a relational database.

You have described at least four entities, and each represents a single idea to be stored as a list. Your six entities are USVs, Payloads, CrewMembers, Qualifications, Missions, and the necessary intersection tables. Each one, like USVs for vessel details or Qualifications for skill types, is a clear, distinct list.

The outline of entity details is thorough. For each one, you describe its purpose, list attributes with datatypes and constraints like primary keys, and describe relationships to other entities. For example, you state that USVs have a one-to-many relationship with Missions.

Your relationships are correctly formulated. The one-to-many relationships, like one USV having many CrewMembers, are clear. You also correctly include at least one many-to-many relationship between USVs and Payloads, which is properly resolved with the USVPayloads intersection table. The design presents a logical view of how all the parts of the fleet management system connect.

There is good consistency in naming. The terms used in the overview match the entity names. A key convention is followed where entity names are plural, like USVs and CrewMembers, and attribute names are singular, like name and serialNumber. The use of capitalization is also consistent throughout the design.

This review and all feedback within it are my own original work. I did not use any AI tools to generate this analysis.

Your draft is excellent and provides a very strong foundation for the project. Well done.

Reviewer: Bryanna Rosales-Hernandez

The problem you are describing is the United Stated Navy tracking personnel assignments and payloads on each vessel by hand. The database website you are proposing will which will allow the commands to have a centralized interface to track the USV status, 150 crew members and qualifications appears to be a great way of solving this issue.

The overview that was a provided gives specific facts of what the solution will provide, for instance the group stated their database website will be able to keep track of *"nearly 150 different mission packages."* Data revolving the issue was also provided, stating *"25 USVs capable of carrying anywhere from one to six payloads,..."*

I believe the group did a great job at detailing the four entities with their intersecting tables provided if the relationship was M:M. The 5 main entities were USVs, Payloads, CrewMembers, Qualifications and Missions.

The outline provided does specify in details the description of each entity, which was similar to how it was described in the overview. For instance, the Payloads entity was described as, *stores the core details and current operational readiness of each Unarmed*

> *Surface Vessel(USV).* The group also described each intersecting table, with it's purpose to ease the M:M relationship.
>
> The group provided two M:M relationships: one between USVs and payload, and another between CrewMembers and Qualifications. The 1:M also appear to be accurately represented in the ERDiagram.
>
> The ERD is consistent in the naming, which makes it easier to read. The attributes are also consistent, it appears that multiple words are used with no spacing but the usage of capitalization helps distinguish words, for example: *serialNumber*.
>
> Feedback was all original, great job on the overview, it was very detailed and the topic is interesting!

Actions based on the feedback

Addressing suggestions from Sy

> One thing to think about is whether CrewMembers should really be limited to a single "primary" USV via a 1:M relationship, since in real operations sailors might rotate or be temporarily assigned elsewhere.

For our specific use case, the 1:M relationship is valid because, for our target command, each USV has a dedicated crew. Other commands who operate smaller USV's than the target audience can have more than 1 USV per crew, but this isn't the target audience. Therefore, no action is necessary.

Addressing suggestions from Charles

> One small suggestion would be to include an example question the system could answer, such as: "Which deployable USVs currently have a sonar payload installed and a qualified operator assigned?"

Solid suggestion. We appended example questions at the overview so our users can better see the use-case for this project.

> One minor thought is that CrewMembers → USVs is modeled as 1:M, but in real life a sailor might work on multiple vessels over time. That could eventually be modeled as M:M

See “Addressing suggestions from Sy” for the same reasoning.

> A small stylistic suggestion would be to rename USVPayloads to USV_Payloads for readability, and maybe shorten CrewMemberQualifications to Crew_Qualifications

As for CrewMemberQualifications > Crew_Qualifications, we actually used this name scheme originally before deciding that it was too ambiguous. The “crew” can refer to multiple people, but the qualifications are assigned to an individual. No action needed.

Addressing suggestions from Jordan

> I think you both covered quite well the generic facts that would be needed/included. Maybe include the potential number of qualifications? Also, maybe considered using different numbers?

Updated sizes of vessels to be accurate to the application. Updated the total crew size to 200. Included statement about this iteration of database only tracking 2 qualifications but in the future it could be updated to contain a limitless number of qualifications.

Addressing suggestions from Mauricio

No suggestions, review was largely affirmative and appreciated! Glad to have the affirmation that we’re on the right track.

Addressing suggestions from Bryanna

No suggestions were offered in this review.

Upgrades to the Draft version

No project file changes necessary. Edited this document’s overview section based on reviewers’ suggestions.

Overview

The United States Navy currently owns an ever growing fleet of approximately 25 Unmanned Surface Vessels (USVs) ranging from small 24-foot reconnaissance craft to medium-sized autonomous vessels roughly 180 ft in length. All of these vessels are modular, meaning that they can be outfitted with one or many payloads or sensor packages. Payloads are either mounted on the back of the vessel or fixed to the mast and include sensors such as a sonar towed array, electronic warfare (EW) suites, and high definition optical sensors. Currently, the personnel assigned to each vessel and the payloads installed on them are all tracked by hand and usually there are many versions of the same file floating around, with minor or major differences. This database-driven website will allow the commands in charge of USVs to have a centralized interface to track USV status (deployed, deployable, training, etc.), which of the 200 crew members are assigned to each vessel, and whether each crew member is qualified or not. Qualifications tracked are limited to USV Operator and USV Supervisor for this application, but could be expanded in the future with the limitless qualifications afforded to the US Navy. With 25 USVs capable of carrying anywhere from one to six payloads, commands can keep track of nearly 150 different mission packages in this database.

Example questions that could be answered (with relative ease) with this project:

- Which deployable USVs currently have a sonar payload installed and a qualified operator assigned?
- Which USV is a particular crew member assigned to?
- When was a particular payload installed on its assigned USV?
- What USVs are currently deployed?
- Is the entire crew of a specified USV fully qualified to operate it?
- What is the vessel class for a USV with a given name?
- What USVs are currently assigned to a mission in a specific location (e.g., "Mediterranean Sea")?
- What is the priorityLevel and location of the mission assigned to a specific USV?

Database Outline

USVs:

Description: Stores the core details and current operational readiness of each Unmanned Surface Vessel (USV).

Attributes:
usvID: int, auto_increment, unique, not NULL, PK
name: varchar(50), not NULL (e.g. “Seahunter”)
class: varchar(50), not NULL (e.g. “MUSV”)
status: varchar(25), not NULL (e.g. “Deployable”, “Maintenance”)

Relationships:
Has an M:M relationship with Payloads via the USVPayloads intersection table.
Has a 1:M relationship with Missions, usvID is a FK in Missions.
Has a 1:M relationship with CrewMembers, usvID is a FK in CrewMembers.

Payloads:

Description: Tracks modular sensors and hardware equipment capable of being mounted on various USVs.

Attributes:
payloadID: int, auto_increment, unique, not NULL, PK
type: varchar(50), not NULL (e.g. "Sonar Towed Array")
serialNumber: varchar(50), unique, not NULL
condition: varchar(25), not NULL (e.g. "Operational")

Relationships:
Has an M:M relationship with USVs via the USVPayloads intersection table.

USVPayloads (Intersection Table):

Description: Facilitates the M:M relationship between USVs and Payloads to track current modular equipment configurations.

Attributes:
usvPayloadID: int, auto_increment, unique, not NULL, PK
usvID: int, not NULL, FK, links to USVs
payloadID: int, not NULL, FK, links to Payloads
installationDate: date, not NULL

CrewMembers:

Description: Records details of the Sailors (Operators and Technicians) responsible for fleet operations.

Attributes:
crewMemberID: int, auto_increment, unique, not NULL, PK
firstName: varchar(50), not NULL
lastName:varchar(50), not NULL
rank: varchar(10), not NULL (e.g. "E6" or "O3")
usvID: int, FK, identifies the primary USV the member is currently assigned to

Relationships:
Has an M:M relationship with Qualifications via the CrewQualifications intersection table.
Linked to USVs via a 1:M relationship.

Qualifications:

Description: Stores a master list of qualifications available to personnel ("USV Operator", "USV Supervisor"

Attributes:
qualificationID: int, auto_increment, unique, not NULL, PK
name: varchar(50), not NULL (e.g. “USV Operator”)

Relationships:
Has an M:M relationship with CrewMembers via the CrewQualifications intersection table.

CrewMemberQualifications (Intersection Table):

Description: Facilitates the M:M relationship between CrewMembers and Qualifications, allowing a Sailor to hold multiple qualifications.

Attributes:
crewMemberQualificationID: int, auto_increment, unique, not NULL, PK
crewMemberID: int, not NULL, FK; links to CrewMembers
qualificationID: int, not NULL, FK; links to Qualifications.
earnedDate: date, not NULL

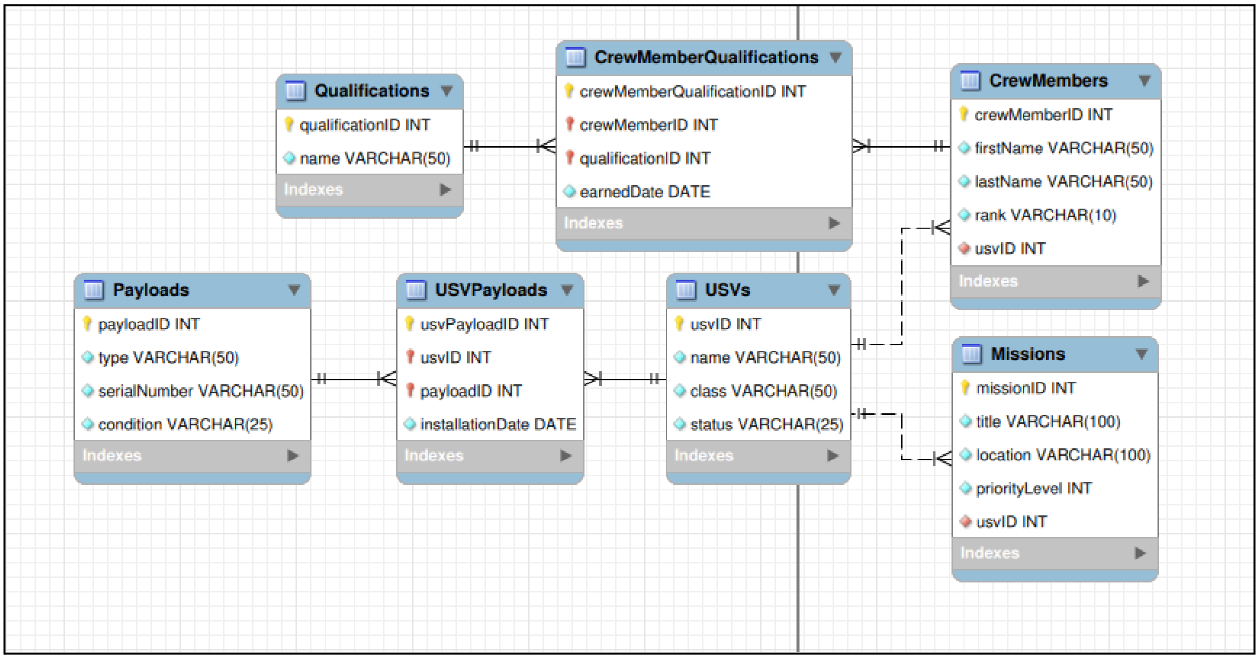
Missions:

Description: Records operational objectives, locations, and priority for USV deployment.

Attributes:
missionID: int, auto_increment, unique, not NULL, PK
title: varchar(100), not NULL (e.g., 'Operation Ghost Net')
location: varchar(100), not NULL (e.g. “Southern California”, “Mediterranean Sea”)
priorityLevel: int, not NULL
usvID: int, FK; identifies which USV is assigned to the task.

Relationships:
Linked to USVs via a 1:M relationship.

Entity Relationship Diagram


Qualifications
qualificationID INT
name VARCHAR(50)
Indexes
CrewMemberQualifications
crewMemberQualificationID INT
crewMemberID INT
qualificationID INT
earnedDate DATE
Indexes
CrewMembers
crewMemberID INT
firstName VARCHAR(50)
lastName VARCHAR(50)
rank VARCHAR(10)
usvID INT
Indexes
Payloads
payloadID INT
type VARCHAR(50)
serialNumber VARCHAR(50)
condition VARCHAR(25)
Indexes
USVPayloads
usvPayloadID INT
usvID INT
payloadID INT
installationDate DATE
Indexes
USVs
usvID INT
name VARCHAR(50)
class VARCHAR(50)
status VARCHAR(25)
Indexes
Missions
missionID INT
title VARCHAR(100)
location VARCHAR(100)
priorityLevel INT
usvID INT
Indexes