Kyle Bell
Jeremy Tsang

Project Step 3 Draft Version: Design HTML Interface

(A) Project Outline and Database Outline

Project Title: Empire Employees
Team: Rakghoul Serum
URL: http://web.engr.oregonstate.edu/~tsangj/340/osu-cs-340-project/empire-employees/

Without proper organization and resolve the galaxy will turn to chaos, or worse, become under the control of rebel scum. Empire Employees manages over 1 billion Stormtroopers along with supporting droids and vehicles. To deploy a specific number of troops to ensure effective defense of each garrison spread such large cosmic distances it is critical to always have the correct number of supplies and personnel available. Empire Employees does this by allowing the users current troop, ship, and droid counts for each garrison and providing sufficient warnings when said counts drop below an acceptable level needed for adequate defense. For some ships, such as tie fighters, there may only be 1 trooper on board whereas for larger ships, such as star destroyers, may have as much as 9,700 Stormtroopers. With our simple to use online interface, we will meet all your Stormtrooper relocation management needs!

Entity Tables:

troopers - an individual soldier of the Galactic Empire that upholds peace and prosperity for all. They have a garrison and load out assignments that are related to the garrisons and loadouts entities respectively.

- id: int, auto_increment, unique, not NULL, PK
- garrison: int, FK
- loadout: int, not NULL, FK
- Relationship: a 1:M relationship between garrisons and troopers, a M:M relationship between troopers and ships (implemented with the ships_troopers composite entity), and a 1:M relationship between loadouts and troopers.

ships - The Galactic Empire has ships in order to transport troopers and droids throughout the galax. Each ship has a ship type and is related to troopers and droids.

- id: int, auto_increment, unique, not NULL, PK
- type: varchar(255), not NULL
- Relationship: a M:M relationship between ships and troopers, a M:M relationship between ships and droids.

droids - Robust, top of the line droids help repair ships for our glorious fleet. They have a droid type, and have a relationship with ships.

- id: int, auto_increment, unique, not NULL, PK
- type: varchar(255), not NULL

Kyle Bell
Jeremy Tsang

- Relationship: a M:M relationship between droids and ships implemented with the ships_droids composite entity.

loadouts - Each unique weapon loadout contains a proprietary blaster as well as an option for a detonator in order for each trooper to get the job done right their own way! A loadout has a relationship with a trooper.

- id: int, auto_increment, unique, not NULL, PK
- blaster: varchar(255), not NULL
- detonator: bool, not NULL
- Relationship: a 1:M relationship between loadouts and trooper

garrisons - Our Empire keeps the galaxy safe with garrisons throughout the galaxy. Each of them has a name and a maximum capacity

- id: int, auto_increment, unique, not NULL, PK
- Name: varchar(255), not NULL
- capacity: int, not NULL
- Relationship: a 1:M relationship is implemented between troops and garrisons as a FK inside of troops.

Relationship Tables:

ships_troopers - Relationship between ships and troopers.

- ship: int, not NULL, PK, FK
- trooper: int, not NULL, PK, FK
- Relationship: facilitates M:M relationship between ships and troopers.

ships_droids - Relationship between ships and droids.

- ship: int, not NULL, PK, FK
- droid: int, not NULL, PK, FK
- Relationship: facilitates M:M relationship between ships and droids.

Team Assignments:

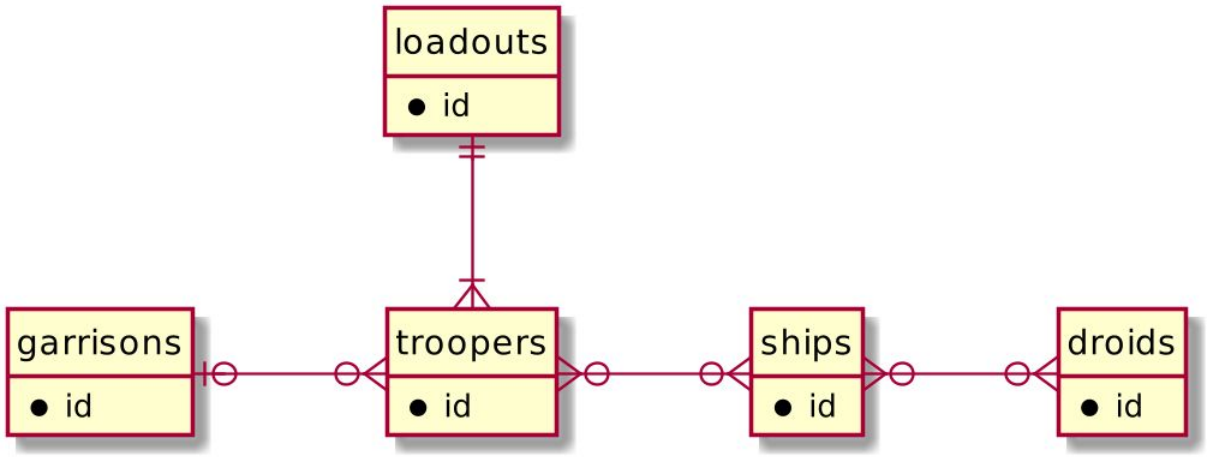
We will be implementing all of the entities described above:

- troopers
- loadouts
- garrisons
- ships
- droids
- ships_droids
- ships_troopers

- Kyle will be responsible for the code and webpages for the following entities:

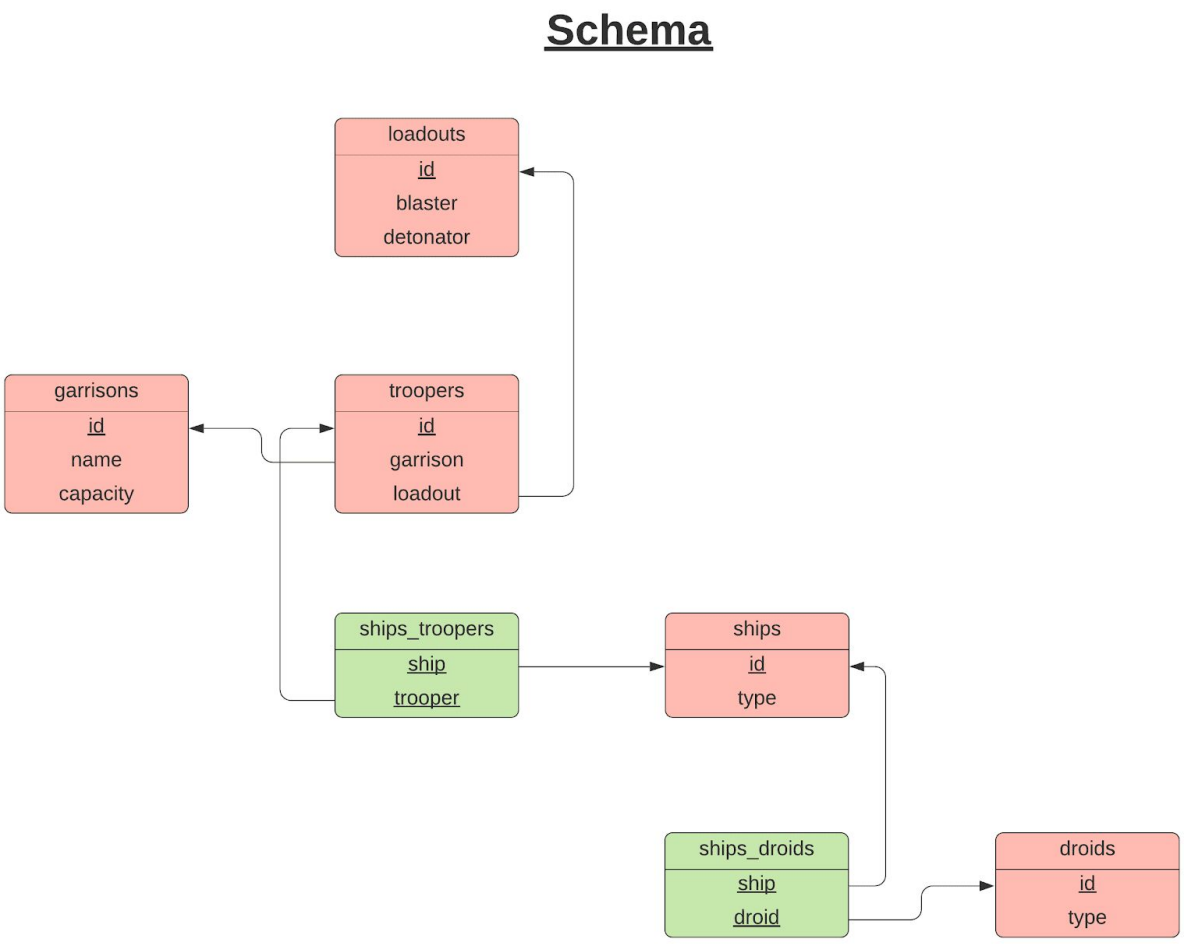
    - ships_droids (M:M relationship)
    - droids
    - garrisons
    - ships
- Jeremy will be responsible for code and webpages for the following entities:
    - ships_troopers (M:M relationship)
    - troopers
    - loadouts

(C) Entity-Relationship Diagram



Kyle Bell
Jeremy Tsang

(D) Schema



Kyle Bell
Jeremy Tsang

(B) Fixes based on Feedback from Previous Steps

Actions taken based on feedback from submitting Step 2 Final Draft

No feedback was received from grades after submitting Step 2 Final Draft

Feedback by the peer reviewers

Jordan Pemberton

Does the overview describe what problem is to be solved by a website with DB back end?

Yes.

Does the overview list specific facts?

Yes, stormtrooper numbers, different numbers of troopers /ship type, etc.

Are at least four entities described and does each one represent a single idea to be stored as a list?

Yes. Could maybe include blasters /detonator as separate entities from loadouts.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

Yes.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

Yes.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

Kyle Bell
Jeremy Tsang

Yes, very consistent.

Kyle Bell
Jeremy Tsang

Rachel Orrell

Does the overview describe what problem is to be solved by a website with DB back end?

Yes, and the idea is very fun! Love it!

Does the overview list specific facts?

Yes, the number of Stormtroopers managed and the number of troopers that can be supported by certain ships.

Are at least four entities described and does each one represent a single idea to be stored as a list?

There are five entities, and I think so, though I don't really understand what a loadout is.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

The purpose of a loadout is not really clear. It says it helps troopers get the job done right - what job? Is it a weapon? I'm just not sure. Also, the description makes it sound like 1:1 ("*a* loadout has a relationship with *a* trooper") while the relationship is listed as M:M. Looks like this was recently changed to 1:M, so neither one really fits. Finally, at least how it is now, shouldn't loadout be specified as an FK?

Otherwise looks good!

It is clear who is responsible for what.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

The implementation of the 1:M for loadouts and troopers makes it so that one loadout can belong to many troopers. Since I don't really understand what a loadout is, I could be off base on this, but if it is a kind of weapon, wouldn't it be more logical for a trooper to have multiple loadouts (the trooper id in the loadout table instead of the loadout id in the trooper table)?

There are 2 M:M relationships, and they are correctly formulated.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

Kyle Bell
Jeremy Tsang

Almost. The foreign keys are referred to as something_id in the relationship tables, but foreign keys in entity tables are just referred to by the singular of the name of the reference table. Having all foreign keys end in "_id" would make it more obvious when an attribute is a foreign key and would improve consistency.

Notes: you don't actually need the crow's feet notation in the schema, and trying to put it there is really confusing when it comes to relationship tables. For example, there will likely be more than just one trooper in ships_troopers, but the crow's feet make it look like exactly one.

Dylan Albertazzi

Does the overview describe what problem is to be solved by a website with DB back end?
Great, you stated the problem, how a DB will help, and some specific details. I would reword the last sentence of your outline paragraph, or split it into two sentences.

Does the overview list specific facts?

Good job listing the specific number of Stormtroopers.

Are at least four entities described and does each one represent a single idea to be stored a s a list?

Yes, five entities are described. I would encourage you to think about more data on each entity that would be useful for your users to have. Such as date the ship was bought or an hourly wage of a trooper.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

Yes, you did a good job being brief but descriptive. I appreciate that. Your descriptions sound a bit like a sales pitch, which is fine, but you’re selling the software, not the entities.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

Yes, all relationships are formulated correctly. Good job.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

All naming is consistent with. Good job.

Kyle Bell
Jeremy Tsang

Patrick Dougan

1. The overview describes the challenges of managing a galactic army. It also explains how the DB will assist with managing troop counts and checking what ships troops are stationed on.

2. The overview lists a couple of facts. One is that total troop count is over 1 billion. Two is star destroyers hold between 1 and 9,700 troops.

3. Yes, they have five entities. Troopers, ships, droids, load outs, and garrisons. Each entity can be described by a list.

4. All entities are described with datatypes and constraints. Relationships are described as well.

5. There are two M:M relationships. The 1:M are correctly formulated

6. Yes, all entities are plural. The "Name" attribute is capitalized in the entity tables list but is OK in the schema.

Actions based on the peer feedback

- Per Rachel Orrell's suggestion, updated the schema reviewer to simply use arrows instead of crow's feet notation. This makes it a bit more streamlined and gets rid of extra information that is included in the ERD anyways.
- Per Rachel Orrell's suggestion, specified that the loadout is a weapon loadout in the outline to be clear that the loadout is weaponry
- Per Rachel Orrell's suggestion, fixed the trooper loadout relationship to consistently be 1:M. This is in accordance with the fact that a loadout can be shared among troopers.
- Per Rachel Orrell's suggestion, reworded/cut down the last outline sentence for better readability
- Per Rachel Orrell's suggestion, removed "_id" from ships_id, trooper_id, and droid_id for consistency and readability.

Actions not taken based on the peer feedback

- We did not implement separate entities for blasters and detonators as suggested by Jordan Pemberton. Those are implemented by our sister company "Empire Weapons."
- We did not include more attributes for each entity (suggested by Dylan Albertazzi) at this time to ensure we have a base database that functions well. As the project progresses we will consider adding more attributes.

Upgrades to the Draft version

No upgrades made on our own after submitting the draft. Upgrades based on the peer comments were sufficient and our database was already in 3NF.

Kyle Bell
Jeremy Tsang

Fixes based on Feedback from Step 1

We did not receive any feedback but we made the following changes:

- Switch naming conventions from camel case to underscores.
- Rename the composite entities shipTrooperManifest and shipDroidManifest to ships_troopers and ships_droids respectively.
- Change loadOuts to loadouts since loadout is a single word.
- Change the relationship between loadouts and troopers from M:M to M:1.
- In the Database Outline for ship_troopers indicate that together ship and trooper form the primary key for composite entity ship_troopers.
- In the Database outline for ship_droids indicate that together ship and droid form the primary key for composite entity ship_droids.