

Justin Phillips  
John Walbert  
CS340 fall 2019  
12/05/19

# **Portfolio**

## **Website Functionality URL**

<http://people.oregonstate.edu/~walbertj/>

## **Executive Summary**

Looking back at all the feedback, the initial proposal and where we are at today we made quite a few changes. In the initial proposal there were only four entities, which meant only 3 relationships were defined. The Employee entity was added to allow for a fourth relationship. The not having enough entities information was pointed out in feedback from a peer. We also decided that we had to add more attributes to the Check Out entity so we added check out date, card number, and employee ID, some of these attributes became foreign keys. The one attribute check out date did not become a foreign key and was just used to track what date they were checked out. We also added another primary key, Employee ID this primary key is used to link employee to checkout which tracks what employee handled the checkout. We also had to add a few more foreign keys book ID and movie ID so that we could link the desired item to checkout. For a more detailed breakdown per step based on the feedback from peers.

In the step 2 draft we needed to add a relationship table to present a many-to-many relationship between checkout and books and checkout and movies. We fixed this by adding the cust\_books and cust\_movies tables. It also pointed out that the data type for phone number should be varchar instead of an int and that our varchar data types should have a number of characters, so we added the (255) to them.

The feedback we received from step 2 final from our peers and the grader was we needed to add a checkout date to the checkout entity. We also needed to add another entity since we were only at 3 relationships. The entity we added was employee with a one-many relationship with checkout. In our schema the grader requested that we change our color since it was hard to see with a white background, so we changed the schema lines to blue. We also needed to incorporate arrows to meet the proper notation requirements. It was pointed out that we had some redundancy with a checkOut attribute where it appeared in movie and cust\_movie. We also had the same issue with cardNum where it was in cust\_book, cust\_movie and in checkout.

In the step 3 review there was a suggestion to help with the HTML UI and we were told that our version of Bootstrap was outdated. The site was updated with the most recent version of Bootcamp 4.

The step 4 peer feedback we received was the main container and the top image shift to the right when going from Home/admin/checkout to books/movies/register. Due to the fact that these pages didn't have a scroll bar on the right so we added the scroll bar to fix this. We also added SQL comments, implemented update for when books and movies are checked out or returned and also implemented delete on customers.

The step 5 feedback we received was that our project did not have a true search/filter and that returning a book or movie wasn't an update, so we added a search feature and the ability to update customers.

## **PROJECT and DB OUTLINES**

The project that we propose is a database that will handle the checkout process for a library. It will allow the library patron to view the available titles, check items out if they are available, and provide a due date for return.

### **Entities**

The first entity that we will have is **customers**. Their attributes will include library card number (a unique identifier), first name, last name and a telephone number. The second entity will be the **checkout**. It is the transaction itself, where the customer borrows books or movies. Its attributes will include transaction ID (a unique identifier), checkout date and due date for return. The third entity will be **books**. Their attributes will include author of the book, the title of the book, a unique ID and if it is already checked out or not. The fourth entity is **movies**. Similar to books, their attributes are title, the year the movie was released, a unique ID and if it is checked out. The last entity is **Employee**. The attributes for it will be employee ID, first name and last name. To support the many to many relationship between books / movies and checkout, the associative entities cust\_movie and cust\_books will be used.

### **Relationships**

The relationships between these entities are defined as follows. For the first entity, the customer, it will have a one to many relationship with the checkout entity. An individual checkout transaction can only be performed by one customer, but an individual customer can have many checkout transactions. The checkout entity will have a many to many relationship with both the books and movies entities. A checkout transaction can contain multiple books and/or movies, as well as each book or movie being associated with multiple checkout transactions. The customer will not have any direct relationship with the movies or the books. The employee entity will have

a one to many relationship with the checkout entity. Many employees can perform checkouts but an individual checkout will only be associated with one employee.

### Constraints

Each entity attribute will have defined data types. For customers the library card number will be an auto-incrementing integer, the first and last names as well as phone number will be varchar(255). The primary key for the customer will be the library card number this will provide a unique identifier for each customer. For the checkout entity, the transaction ID will be an auto-incrementing integer and also the primary key for that entity. It will also server as a foreign key between the checkout entity and the customer to allow tracking of checkout transaction to a particular customer. The due date and check out date will be will be stored as date type. For books and movies, the title will be varchar(255), checked out will be bool data type, and ID will be an auto-incrementing integer (also the primary key for both of those entities). Specifically for movies, the year of release will be stored as an int, and for books the author will be stored as varchar(255). Using these unique attributes the customer will be able to go select the movies and books they wish to checkout from a list that shows what is currently in stock. These items will be added to the checkout then will be assigned a due date and transaction id. After the customer finished the transaction, it will change the checked out attribute for the items that were borrowed from false to true, and will subsequently reflect this on additional searches. There will be an Employee entity that will be linked to check out and handle the check out process. The employee id will be an auto-incrementing integer. The first and last names will be varchar(255) .

### Entities : Attributes

**Books** : *Title, Author, ID, Checked Out*

**Check Out** : *Transaction Num, Due Date, Check Out Date, Card Num, Employee ID*

**Customer** : *Library Card Number, First Name, Last Name, Phone Number*

**Movies** : *Title, Year, ID, Checked Out*

**Employee** : *Employee ID, First Name, Last Name*

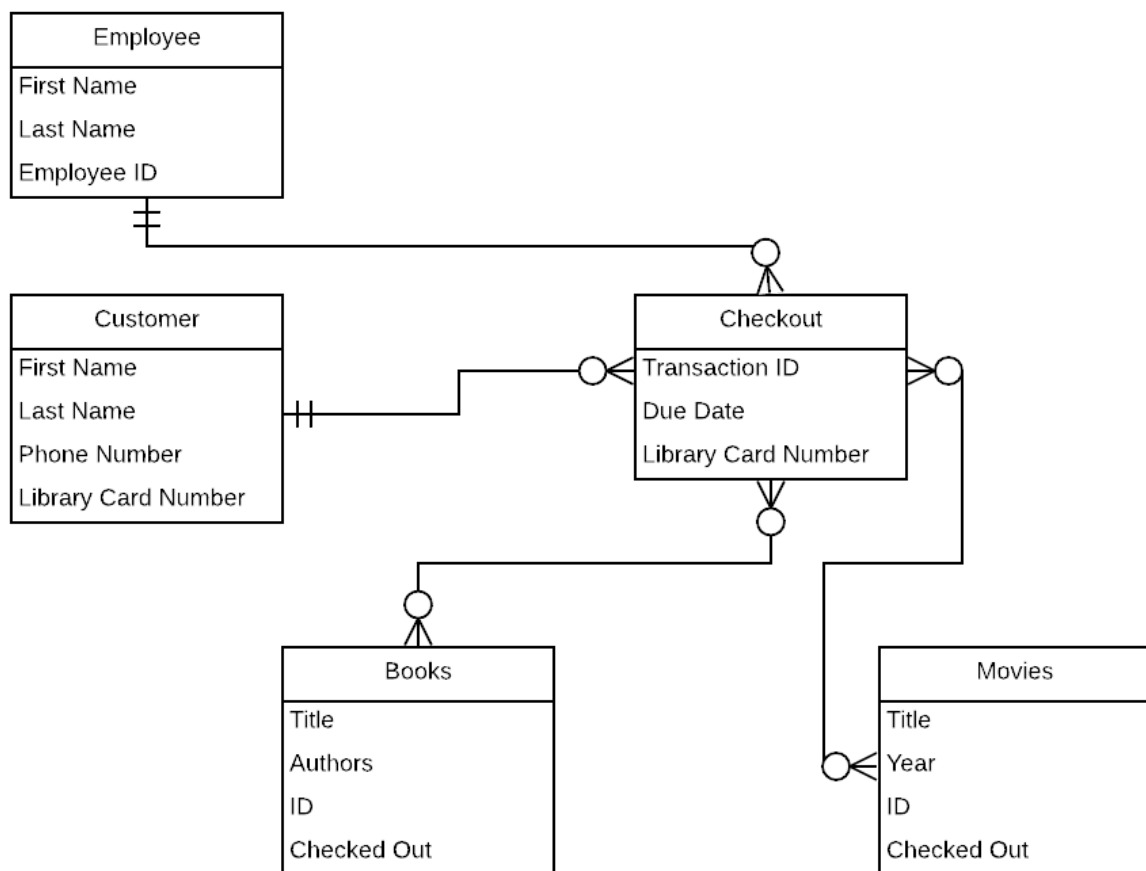
### Constraints : Attributes

**Primary key** : *library card number, transaction ID, ID (books / movies), Employee ID*

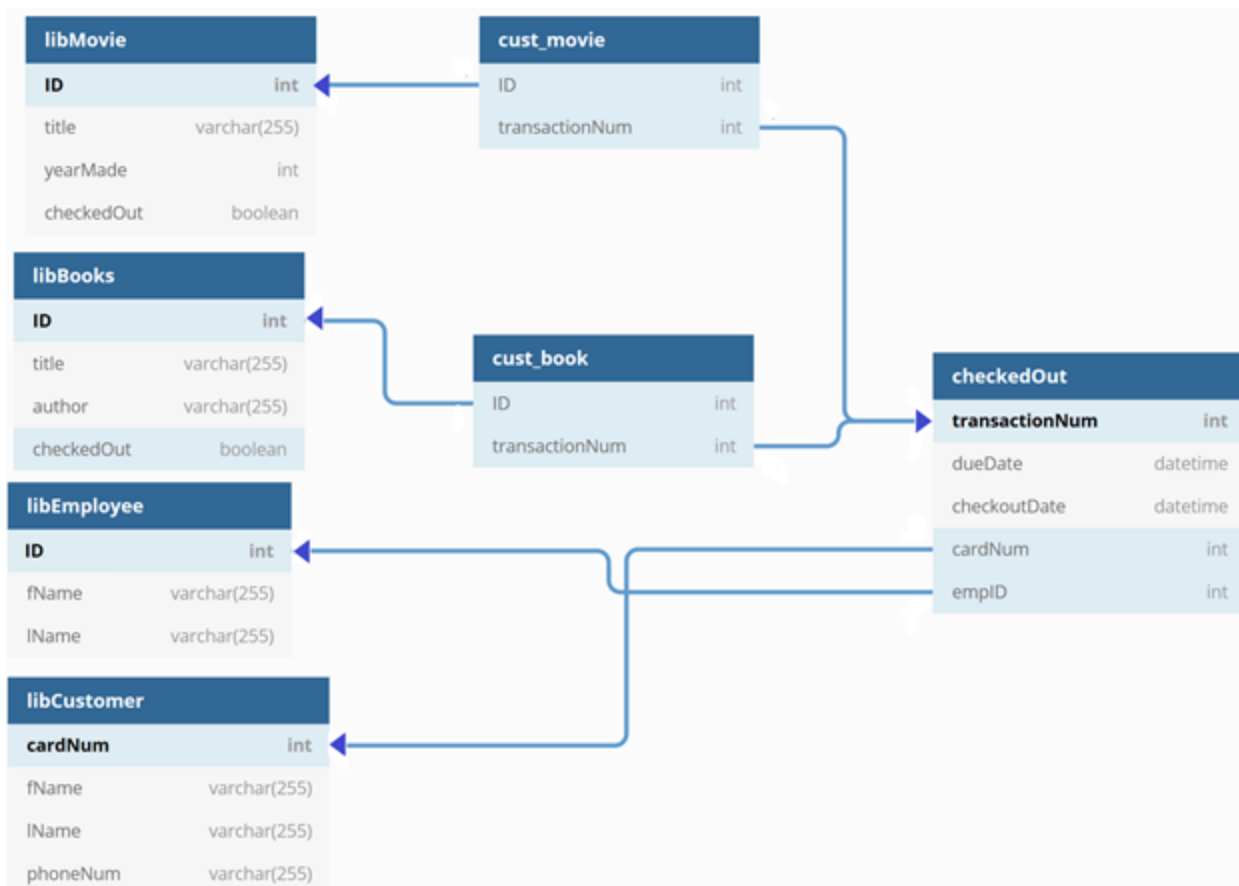
**Not Null** : *All attributes*

**Foreign key** : *transaction ID, Employee ID, ID (books), ID (movies)*

## ER DIAGRAMS

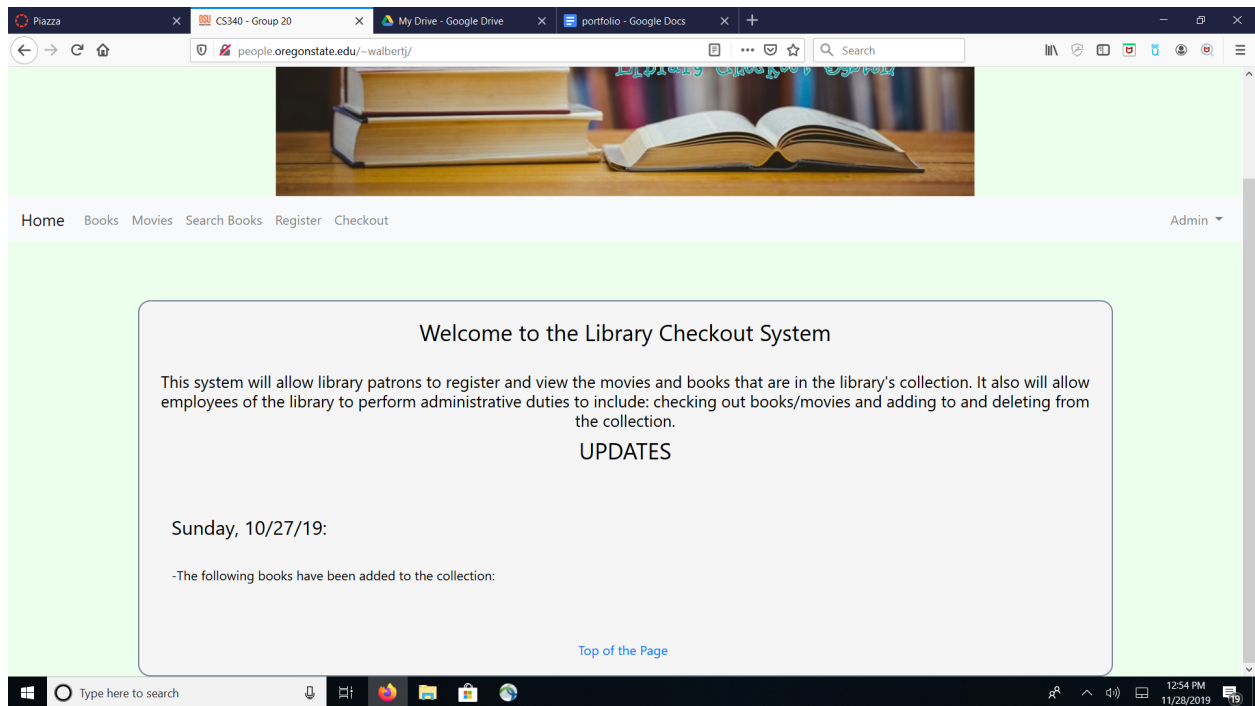


## SCHEMA



## UI Screenshots

### Main Page



## READ All the Books in the Inventory

The screenshot shows a web browser window with multiple tabs. The active tab is 'CS340 - Group 20'. The address bar shows 'people.oregonstate.edu/~walbertj/#'. The web application has a navigation bar with 'Home', 'Books', 'Movies', 'Search Books', 'Register', and 'Checkout'. The 'Books' section is active, displaying a table with the following data:

Title	Author	Available
The Notebook	Nicholas Sparks	No
It	Stephen King	Yes
Cujo	Stephen King	No
Warriors	George R R Martion	Yes
The Lost Symbol	Dan Brown	Yes
Iron Wolf	Dale Brown	No
Orient	Christopher Bollen	Yes
Watchers	Dean Koontz	No
Lonesome Dove	Larry McMurty	No
All the Pretty Horses	Cormac McCarthy	Yes
Super Bad	Seth Rogen	Yes

## READ All Movies in the Inventory

The screenshot shows the same web application with the 'Movies' section active. The table displays the following data:

Title	Year	Available
The Notebook	2004	Yes
It	2018	No
Cujo	1983	No
Titanic	1997	Yes
Star Wars: A New Hope	1977	No
Pulp Fiction	1994	No
Taxi Driver	1976	Yes
The Godfather	1972	No
Forrest Gump	1994	No
Lonesome Dove	1989	No

## READ/SEARCH using a title to find a book

The screenshot shows a web browser window with the URL `people.oregonstate.edu/~walbertj/#`. The page features a header with the title "LCOS Library Checkout System" and a navigation menu with links: Home, Books, Movies, Search Books, Register, and Checkout. An "Admin" link is also present. The main content area contains a "Search" section with a "Book Title:" label, a text input field, and a "Search" button. The browser's taskbar at the bottom shows the Windows logo, a search bar, and several application icons, with the system clock indicating 12:55 PM on 11/28/2019.

LCOS  
Library Checkout System

Home Books Movies Search Books Register Checkout Admin

Search

Book Title:

## CREATE a new Customer

This screenshot displays the "New Customer Registration" form on the same website. The form includes fields for "First Name" (with placeholder text "First"), "Last Name" (with placeholder text "Last"), and "Phone Number" (with placeholder text "111-111-1111"). A "Register" button is located at the bottom left of the form area. The browser window and taskbar are identical to the previous screenshot, showing the same URL and system time.

LCOS  
Library Checkout System

Home Books Movies Search Books Register Checkout Admin

New Customer Registration

First Name:

Last Name:

Phone Number:



## CREATE a Checkout order

Checkout

Book ID:  Add Book

Movie ID:  Add Movie

Cart

Title	Type	Due Date
-------	------	----------

Library Card Number:  Check Library Card

Employee ID:  Check Employee

## CREATE/READ/UPDATE\*/DELETE Books in inventory (\*updates 'Available' when returned)

Add Book

Title:  Author:  Add Book

Current Books

Id	Title	Author	Available	Options
1	The Notebook	Nicholas Sparks	No	<button>Delete</button> <button>Return</button>
2	It	Stephen King	Yes	<button>Delete</button>
3	Cujo	Stephen King	No	<button>Delete</button> <button>Return</button>
4	Warriors	George R R Martion	Yes	<button>Delete</button>
5	The Lost Symbol	Dan Brown	Yes	<button>Delete</button>

## CREATE/READ/UPDATE/DELETE Movies in inventory

Home Books Movies Search Books Register Checkout Admin

### Add Movie

Title:  Year Released:  [Add Movie](#)

### Current Movies

Id	Title	Year	Available	Options
1	The Notebook	2004	Yes	<a href="#">Delete</a>
2	It	2018	Yes	<a href="#">Delete</a>
3	Cujo	1983	No	<a href="#">Delete</a> <a href="#">Return</a>
4	Titanic	1997	Yes	<a href="#">Delete</a>
5	Star Wars: A New Hope	1977	Yes	<a href="#">Delete</a>
6	Pulp Fiction	1994	Yes	<a href="#">Delete</a>

## CREATE/READ/DELETE Employee

Home Books Movies Search Books Register Checkout Admin

### Add Employee

First Name:  Last Name:  [Add Employee](#)

### Current Employees

Id	First Name	Last Name	Options
1	Justin	Phillips	<a href="#">Delete</a>
2	John	Walbert	<a href="#">Delete</a>
3	Johan	Von Beethoven II	<a href="#">Delete</a>

## CREATE/READ/UPDATE/DELETE Customers

The screenshot shows a web browser window with multiple tabs. The active tab is 'CS340 - Group 20' with the URL 'people.oregonstate.edu/~walbertj/'. The page has a navigation bar with links: Home, Books, Movies, Search Books, Register, Checkout, and an Admin dropdown. The main content area is titled 'Current Customers' and contains a table with the following data:

Id	First Name	Last Name	Phone Number	Options
6	Howdy	Partner		<a href="#">Delete</a> <a href="#">Update</a>
8	Pete	The Dragon	999-999-9999	<a href="#">Delete</a> <a href="#">Update</a>
9	Franklin	BBQ	800-888-8888	<a href="#">Delete</a> <a href="#">Update</a>
10	Double	Dragon		<a href="#">Delete</a> <a href="#">Update</a>
11	NewCusto	Mer1	123-867-5309	<a href="#">Delete</a> <a href="#">Update</a>
12	Bill	Bixby	555-555-5555	<a href="#">Delete</a> <a href="#">Update</a>

## READ/DELETE Checkouts

The screenshot shows the same web browser window as before, but the main content area is titled 'List of Checkouts' and contains a table with the following data:

Checkout Id	Library Card Number	Customer Name	Employee Name	Due Date	Options
12	9	Franklin BBQ	John Walbert	2019-12-11	<a href="#">Delete</a>
13	8	Pete The Dragon	John Walbert	2019-12-11	<a href="#">Delete</a>
15	9	Franklin BBQ	Johan Von Beethoven II	2019-12-11	<a href="#">Delete</a>
16	9	Franklin BBQ	Justin Phillips	2019-12-11	<a href="#">Delete</a>
18	8	Pete The Dragon	Johan Von Beethoven II	2019-12-11	<a href="#">Delete</a>
19	8	Pete The Dragon	Justin Phillips	2019-12-11	<a href="#">Delete</a>
21	6	Howdy Partner	John Walbert	2019-12-11	<a href="#">Delete</a>
23	6	Howdy Partner	John Walbert	2019-12-11	<a href="#">Delete</a>