Circuit Collective

Lab 10
Customer Meeting for March 28th

Team Members: William Wedemire, Bryce Gill, Nathan Aguiar, Tobenna Nnaobi, Jerry Yang

Agenda for Lab 10 Meeting

For this meeting we will…

1. Present our proposed testing plan.
2. Allow for customer interaction with build scripts.
3. Present an updated version of the iteration 3 deliverables.
4. Confirm the final feature list.
5. Present an updated iteration 3 feature list.
6. Proposition for a new feature.
7. Present documentation plan and initial documentation.
8. Present updated burndown chart.
9. Present Task Board and Repository.
10. Iteration 2 Feature Demonstration

Testing Plan

Clear Box tests

Test type	Member assigned	Components Involved / Steps	Description / How it will be done	Customer Sign off Y/N
Unit Test	Nathan	insert()	Backend test that checks if new item is successfully added to database or not. The test will check what happen if input is different, such as null, or different item name, price, stock, etc..	Y
Unit Test	Nathan	list()	Backend test that checks function used for getting list of item, the test will check what function will return with different input.	Y
Integration Test	Nathan	Web API on localhost	Checks if web api can be accessed by javascript on frontend with given port, if success, JSON responses depend on method called by fetch() in javascript.	Y
System Test	Nathan	Logging	Check if user successfully logged in with correct username and password, and check is user directed to category page or not.	Y

Translucent Box tests

Test type	Member assigned	Components Involved / Steps	Description / How it will be done	Customer Sign off Y/N
Unit Test	Nathan	search()	Backend test that checks if search function works successfully with different input, if success, function will return index table with given input as search result. Apache Lucene Search Algorithm is used and warped by custom function.	Y
Integration Test	Jerry	Inserting item	Check if item info provided from frontend is being successfully added to database or not.	Y
Integration Test	Jerry	Delete item	Check if item info provided from frontend is being successfully deleted from database or not.	Y
Integration Test	Jerry	Search item	Given input from frontend, checks if search returned from backend is matching/approximate as input or not.	Y
Integration Test	Jerry	Edit item	Given input with what attribute is being edited and what is changed, check if backend have successfully record new changes or not.	Y

Opaque Box tests

Test type	Member assigned	Components Involved / Steps	Description / How it will be done	Customer Sign off Y/N
Unit Test	Tobenna	fetch(\`${api}/game/search?l=25&q=${query}\`)	Calls from javascript, fetch data from backend using search API, if it success, check if result is reasonable or not.	Y
Unit Test	Tobenna	fetch(\`${api}/game\`)	Call from javascript, creating a new item and insert it to database, if it successfully added to database. Result will return and display in category page.	Y
Unit Test	Tobenna	fetch(\`${api}/admin/game/${id}\`, { method: "DELETE"})	Call from javascript, deleting item and removing it from database, if it successfully removed from database. Result will return and display in category page.	Y
Unit Test	Tobenna	fetch(\`${api}/admin/game/${id}\`, { method: "POST"})	Call from javascript, changing attribute of item, if it is successfully changed and recorded in database. Result will return and display in category page.	Y

Opaque Box tests

Test type	Member assigned	Components Involved / Steps	Description / How it will be done	Customer Sign off Y/N
System Test	Jerry	Search bar	Typing string inside search bar, hit enter will filter out category list, only search result will be displayed in list. Different input will be test to check validity	Y
System Test	Jerry	Website category item list editing	In web page, test checks inputting different item and using add to see if item is added into category list or not, use delete button checks if it is deleted from category list or not, and use edit to change item attribute checks if the attribute change is displayed inside category list or not.	Y

Build Script Interaction

Current Feature List

Search Bar Track Stock Login Password	Done Done Done
Game Tag Revenue Tracking	Done Done
Search Bar Recommendations Purchase Logging Shipment Tracking	In Progress In Progress In Progress

Software Iteration Updates

Current Iteration Updates

Game Tag - Done

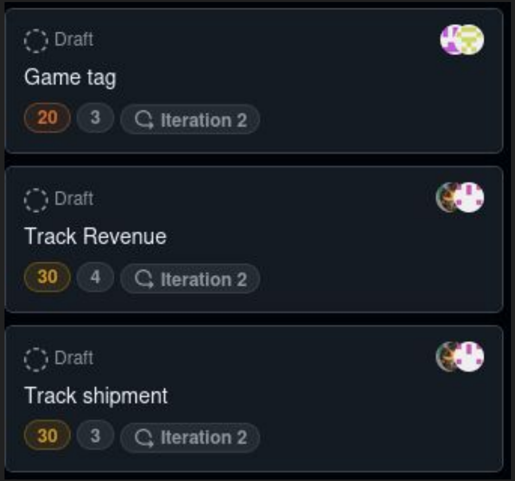
Allow inventory controllers the ability to add tags to games

Priority: 20
Estimated Time: 3 days
Assigned: Jerry, Tobenna

Track Revenue - Done

Allow managers to view data on sales and revenue analytics to relay to customers

Priority: 30
Estimated Time: 4 days
Assigned: Nathan, Bryce, William



Draft
Game tag
20 3 Iteration 2

Draft
Track Revenue
30 4 Iteration 2

Draft
Track shipment
30 3 Iteration 2

Iteration 3

To be completed during before lab 11

Search bar recommendation list

Add functionality to the search bar to give recommendations based on the search

Priority: 30
Estimated Time: 4 days
Assigned: Nathan, Tobenna

Purchase Log

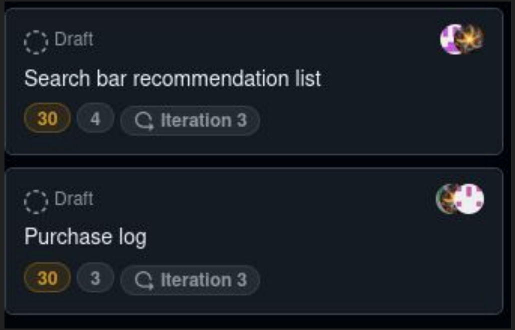
Add a purchase log so managers can manage customer purchase log

Priority: 30
Estimated Time: 3 days
Assigned: Nathan, Bryce, William

Track Shipment - In Progress

Allow managers to keep track of shipments.

Priority: 30
Estimated Time: 3 days
Assigned: Nathan, Bryce, William



Draft
Search bar recommendation list
30 4 Iteration 3

Draft
Purchase log
30 3 Iteration 3

Any Desired Changes For Iteration 3?

Search Bar Recommendation List
Purchase Log
Track Shipments

Documentation Plan

Initial Documentation:

Readme and documentation folder in Github.

Developer Documentation:

Lead: Bryce

Consultation: Nathan

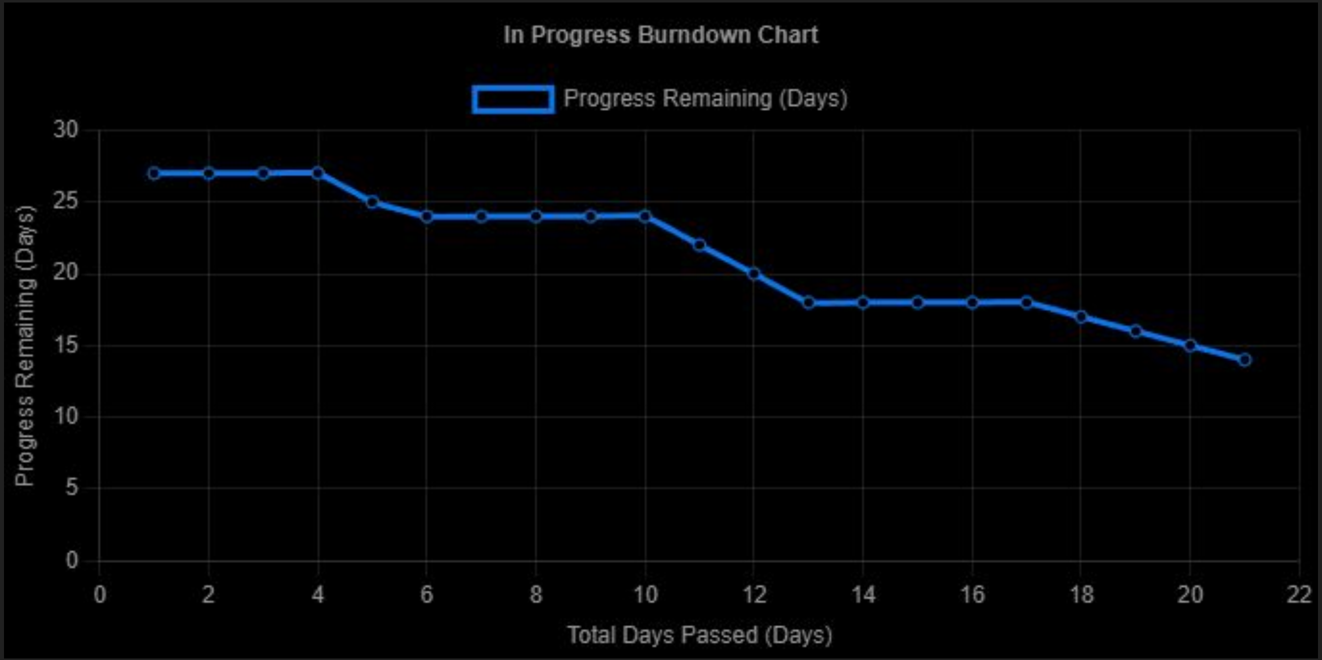
User/In-application Documentation

Lead: Bryce

Consultation: Tobenna

All documentation will be saved as PDFs. User Documentation will be distributed with application.

Burn-down chart

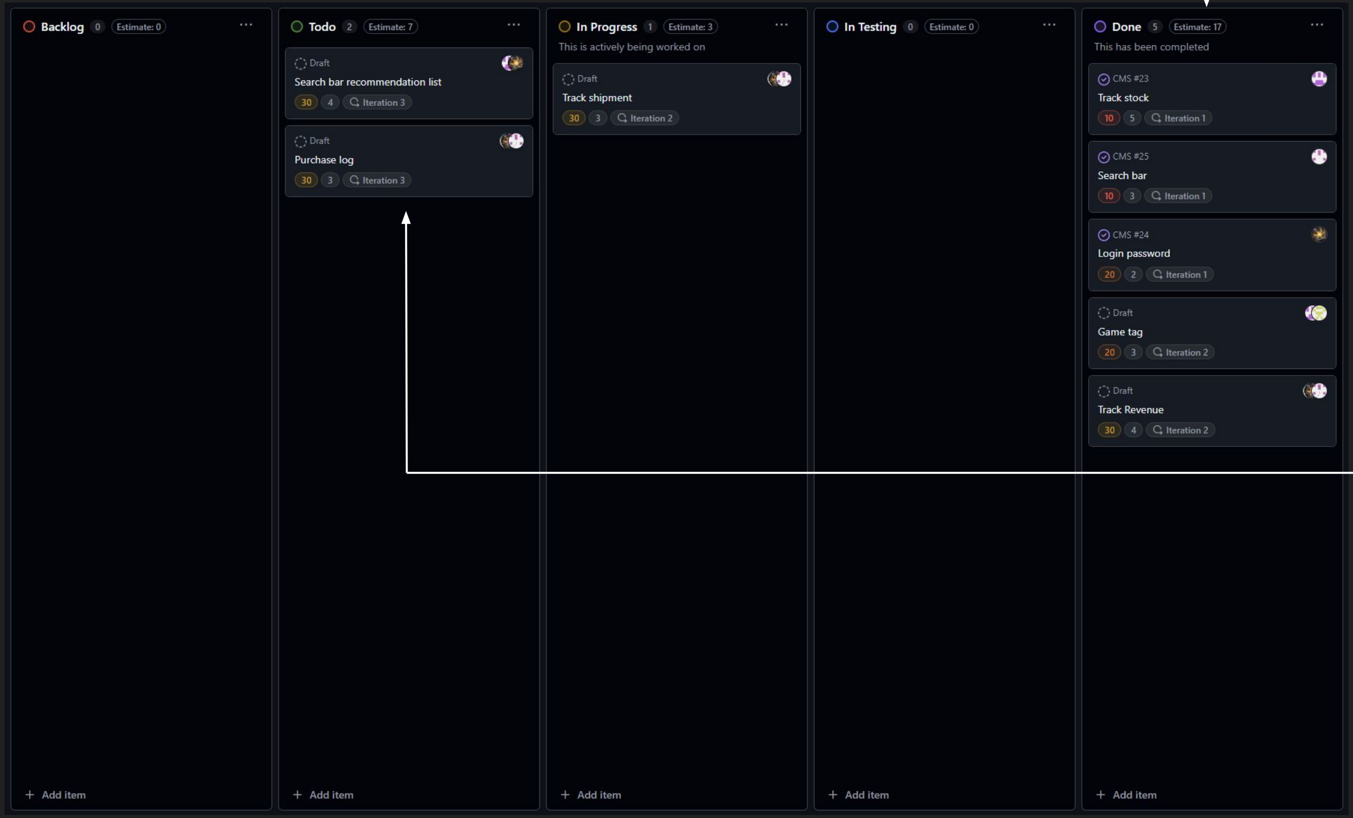


Progress:

Game Tagging - Done

Track Shipment - 2 Days

Task Board



Backlog 0 Estimate: 0

Add item

Todo 2 Estimate: 7

Draft
Search bar recommendation list
30 4 Iteration 3

Draft
Purchase log
30 3 Iteration 3

Add item

In Progress 1 Estimate: 3
This is actively being worked on

Draft
Track shipment
30 3 Iteration 2

Add item

In Testing 0 Estimate: 0

Add item

Done 5 Estimate: 17
This has been completed

CMS #23
Track stock
10 5 Iteration 1

CMS #25
Search bar
10 3 Iteration 1

CMS #24
Login password
20 2 Iteration 1

Draft
Game tag
20 3 Iteration 2

Draft
Track Revenue
30 4 Iteration 2

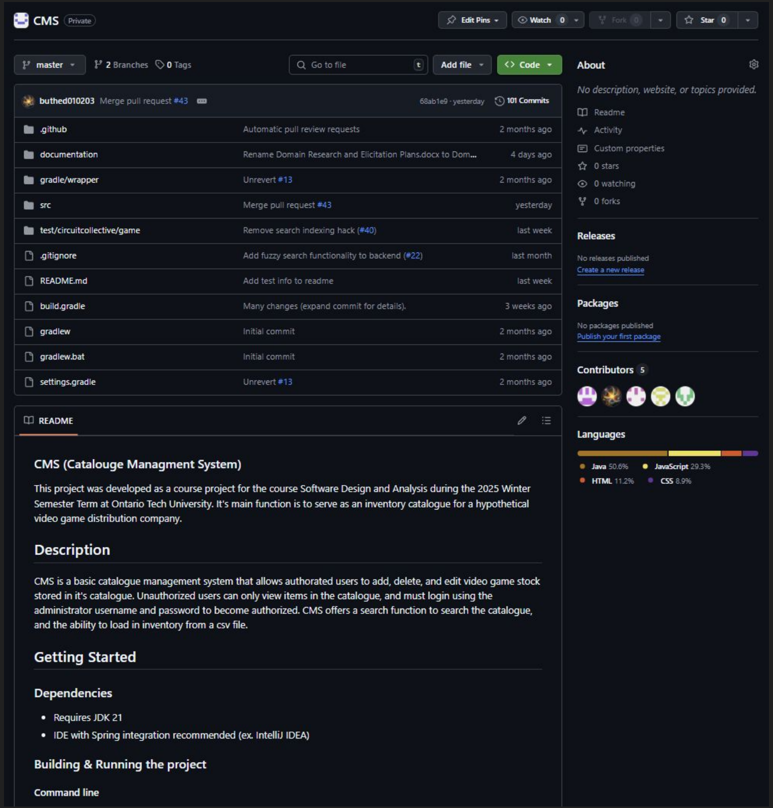
Add item

Since Last Meeting

Finished
Revenue Tracking

Currently in Progress
Search Bar Recommendations
Purchase Logging

Repo screenshot



buthed010203 Merge pull request #43 68ab1e9 · yesterday 101 Commits

.github	Automatic pull review requests	2 months ago
documentation	Rename Domain Research and Elicitation Plans.docx to Dom...	4 days ago
gradle/wrapper	Unrevert #13	2 months ago
src	Merge pull request #43	yesterday
test/circuitcollective/game	Remove search indexing hack (#40)	last week
.gitignore	Add fuzzy search functionality to backend (#22)	last month
README.md	Add test info to readme	last week
build.gradle	Many changes (expand commit for details).	3 weeks ago
gradlew	Initial commit	2 months ago
gradlew.bat	Initial commit	2 months ago
settings.gradle	Unrevert #13	2 months ago

README

CMS (Catalouge Managment System)

This project was developed as a course project for the course Software Design and Analysis during the 2025 Winter Semester Term at Ontario Tech University. It's main function is to serve as an inventory catalogue for a hypothetical video game distribution company.

Description

CMS is a basic catalogue management system that allows authorated users to add, delete, and edit video game stock stored in it's catalogue. Unauthorized users can only view items in the catalogue, and must login using the administrator username and password to become authorized. CMS offers a search function to search the catalogue, and the ability to load in inventory from a csv file.

Getting Started

Dependencies

- Requires JDK 21
- IDE with Spring integration recommended (ex. IntelliJ IDEA)

Building & Running the project

Command line

About

No description, website, or topics provided.

Readme
Activity
Custom properties
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 5

Languages

● Java 50.6% ● JavaScript 29.3%
● HTML 11.2% ● CSS 8.9%

https://github.com/CircuitCollective/CMS

Iteration 2 Feature Demonstration

Agenda for Lab 11 Meeting

For the next meeting…

1. Present final plans to polish the project and deliver a functional product.
2. Discuss any remaining features.
3. Discuss plans for documentation.
4. Ensure the project is on the correct path.
5. Present final updated task board and burn down chart.
6. Present final demonstration.
7. Present results of testing.
8. Present complete project retrospective.

Questions & Feedback