Innovation Hub Inventory Management

By

Group 5

Shreya Srivastava

Ramya Chandrasekhar

Mohan Suraj Lakshmi Narayana

Venkata Siva Abhishek Munukutla

Aditya Kalyanaraman

Team Project Report Presented in Partial Fulfillment
of the Requirements for the Course Project
IFT 540

Approved December 2018 by the
Graduate Supervisory Committee:
Dr. Joseph Kuitche

ARIZONA STATE UNIVERSITY

December 2018

ABSTRACT

The objective of the Innovation Hub Inventory Management project is to create a online system that enable the staff at Innovation Hub and students/faculty from ASU to purchase materials and schedule printers for 3D printing. The current system has too much manual work involved. We propose a system with automated workflows. The features, structure and contents of the system have been developed based on the requirements primarily gathered from industry standards documents, interviews and research papers. By providing users with a platform to place order for purchasing materials and providing the staff with a system that does not rely heavily on manual input, the application aims to help Innovation Hub transition from a totally manual system to a system with automated workflows.

ACKNOWLEDGEMENTS

This project report would not have been possible without the support of our instructor and mentor, Dr. Joseph Kuitche. We thank the Instrument Shop Supervisors of Arizona State

University's Innovation Hub for their guidance. We also thank our colleagues from Arizona State University's Information Technology Department, whose valuable feedback helped improve this report as well as the project's implementation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

ASU’s Innovation Hub provides students the room and tools they need to prototype and build creative projects and features equipment that includes a ShopBot 3D wood router, Epilog Laser engraver and 3-D printers. Managing and scheduling of resources is done manually using spreadsheets. As the current system is entirely managed manually, we are working to make an automated application which will replace the manual process thereby increasing the productivity by more than 80% and will also decrease the error rate to a great extent.

Our goal is to help Innovation Hub in transitioning from manual resource management and scheduling system to a system with automated workflows.

We mainly focussed on building the Inventory Management module for the Innovation Hub. Due to time crunch, we had to skip over on machine utilization and lab management and the quoting system as the entire project would definitely take more time than what was available to us throughout the course of this project.

This report has been structured into different sections based on the life cycle of the project. The different sections are:

1. Requirements Gathering
2. Structuring Requirements
3. Structuring Data Requirements
4. User Interface
5. Implementation

SECTION 1: REQUIREMENTS GATHERING

The requirement gathering process undertaken for our project primarily involved the use of a couple of fact-finding techniques viz., interviews and document analysis. The first step in this process was to analyze various technical documents available which would give us a brief idea regarding this project. This involved reaching out to the supervisors at Innovation Hub and setting up meetings with them in order to gather and understand the requirements. After completing the document analysis, an interview session was conducted in order to better understand the end-user requirements for this information system and resolve any discrepancies regarding the ideas put forth. The final outcome of the requirements gathering phase was formulating the Requirements Definition Statement and clearly defining the functional and non-functional requirements of the system. Details of this phase are given below:

Document Analysis

This process involved studying the documents provided to us in order to determine the system requirements. The documents provided to us were the Project Team Purchase and the Personal Purchase Excel spreadsheets and the Innovation Hub user manual. The main outcome of this process was identifying various entities, their attributes and relationships that were to be included in the corresponding data modeling phase. Important parameters such as the kind of project being worked on (Course Project, Personal), the type of printer to be requested, the type of parts required and the cost per measure of each part were recognized.

During this process, our main strategy or approach was to focus on the technical specifications for performing the documents analysis. The information derived from these documents were primarily about the various technical parameters pertaining to the manual Inventory Management system being used by the Innovation Hub. This helped tremendously in modelling and designing the database for the system.

The documents used in this phase have been provided in the References section of this report.

Interviews

During this process, interview sessions with Mr. Sean Dengler, who is the Instrument Shop Supervisor of the Innovation Hub and later with Mr. Derek Robinson were conducted. This was done to better understand the system as expected by the end-user of the product, and to resolve any differences concerning the proposed implementation. The interview process was vital to establishing a list of project requirements. Prior to these interviews, the only resources we had to come up with the requirements were a set of documents. While these documents provided insight into the testing process and the data that our system will be handling, it mainly helped us bridge this gap between the technical and practical aspects of the project. The interview also clarified the scope of the project, thereby clearly defining the features that needed to be implemented in the final product.

The interview report, which contained the minutes of the meeting, has been provided in the Appendix section of this report.

Requirements Definition Statement

Based on the document analysis and the interviews conducted, we formulated a Requirements Definition Statement, given below:

"A interface and a database needed to be developed for facilitating the inventory management. Different login authorizations needed to be provided to several users of the system with respect to their position in Arizona State University."

Also, in this phase, we jotted down a list of functional and nonfunctional requirements that would be a part of the system. The list is as follows:

1. Functional Requirements
    - 1.1. Process-Oriented
        - 1.1.1. Ability to request for materials
        - 1.1.2. Adding new materials to the inventory
        - 1.1.3. Update the existing inventory
        - 1.1.4. Ability to view inventory
    - 1.2. Information-Oriented
        - 1.2.1. Maintaining a list of materials in the inventory

2. Non-Functional Requirements
    - 2.1. Operational
        - 2.1.1. System should be compatible with majority of web browsers
    - 2.2. Performance
        - 2.2.1. System should be able to handle multiple order requests at the same time

2.3. Security

2.3.1. System requires login for authentication

2.3.2. Providing access according to user roles

2.4. Cultural and Political

2.4.1. System must be in compliance with local laws and regulations pertaining to information privacy and other content.

SECTION 2: STRUCTURING REQUIREMENTS

Features of the system

Based on observations from the previous phase, a list was prepared consisting of all the features to be included in the system. They are mentioned below:

1. User Login

a. Login Form

i. Student/Faculty login

ii. Administrator login

2. View Dashboard

a. View current status of materials in inventory

b. Update form

3. Inventory

a. Updating inventory

i. Selecting item to update

ii. Update form

4. Update Menu

a. View Item List

i. Adding Item

1. Form for adding item

ii. Select Item

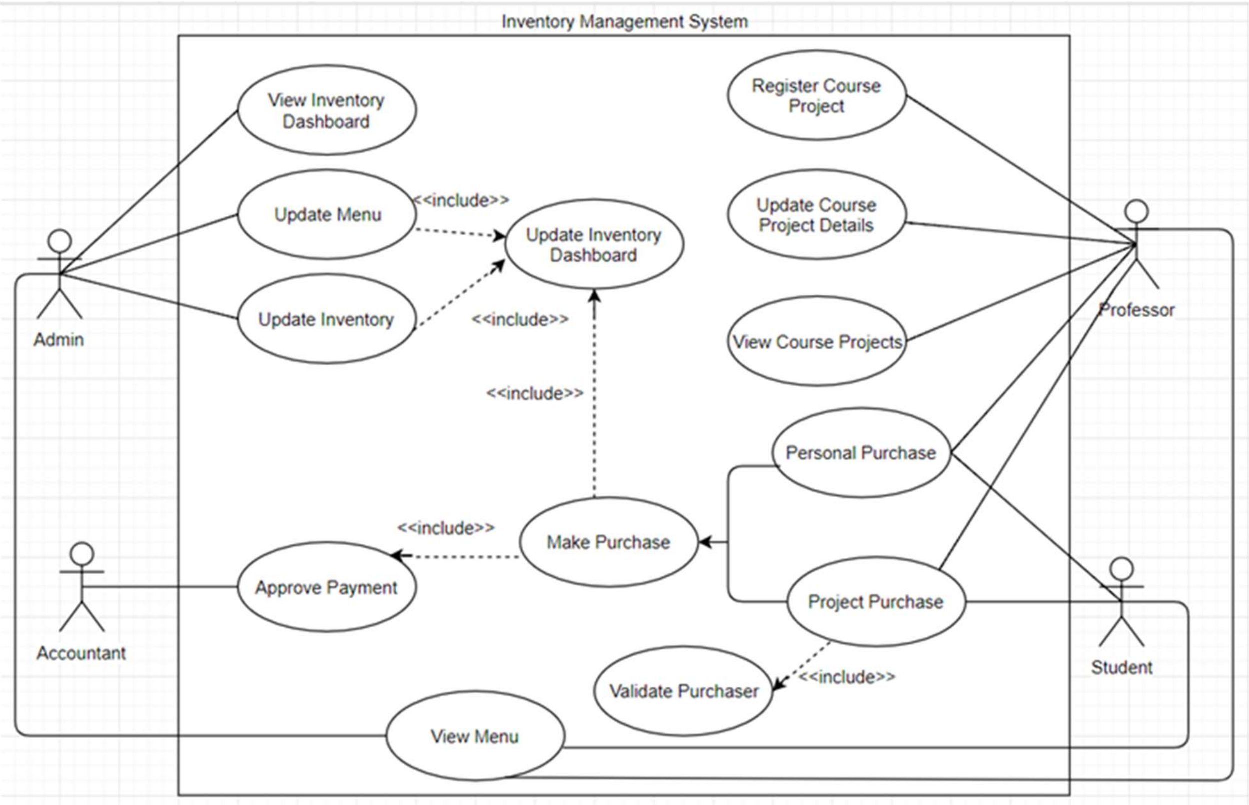
1. Deleting item

5. Student Login

a. Make purchase

Use Case Diagram

After finalizing the list of features to be included in the final system, various actors, use cases and system boundaries were identified. Following this, a use case diagram was created depicting the actors and their interactions with the system. The use case diagram is shown below:



Use Case Diagram for Inventory Management

Use Case Descriptions

As a part of the Structuring Requirements phase, use case descriptions were written. They are listed below:

USE CASE 1: View inventory dashboard

Use Case ID:	1
Use Case Name:	View inventory dashboard

Created By:		Last Updated By:	
Date Created:	10/22/2018	Date Last Updated:	10/22/2018

Actors:	Administrator
Description:	This use case allows administrator to view the inventory dashboard.
Trigger:	Administrator opens up the inventory management application.
Preconditions:	1. Administrator must be authenticated. 2. Inventory dashboard must have items to view.
Postconditions:	Items in the inventory are displayed to view.
Normal Flow:	1. Administrator enters the inventory management application and clicks View Dashboard button. 2. The dashboard is displayed with list of items along with quantity.
Alternative Flows:	2.1 1. The dashboard does not have items to display. 2. The user is redirected to add items page.
Exceptions:	
Assumptions:	Administrator has an internet connection to access the application.
Notes and Issues:	

USE CASE 2: Update inventory

Use Case ID:	2		
Use Case Name:	Update inventory		
Created By:		Last Updated By:	
Date Created:	10/22/2018	Date Last Updated:	10/22/2018

Actors:	Administrator
Description:	This use case allows administrator to update the inventory.
Trigger:	Administrator clicks the update inventory button.
Preconditions:	1. The administrator is required to add or change items in the inventory.
Postconditions:	1. Items in the inventory are updated.
Normal Flow:	1. Administrator clicks the Update Inventory button. 2. The page for updating the items loads up. 3. The admin makes the required changes and clicks on Save button. 4. The inventory is updated.
Alternative Flows:	2.1 1. The update items page does not load. 2. The admin refreshes the browser. 3.1 1. The save action does not go through. 2. Inventory is not updated. 3. The update inventory page is refreshed.
Exceptions:	

Assumptions:	Administrator has an internet connection to access the application.
Notes and Issues:	

USE CASE 3: View Menu

Use Case ID:	3		
Use Case Name:	View Menu		
Created By:		Last Updated By:	
Date Created:	10/22/2018	Date Last Updated:	10/22/2018

Actors:	Administrator, Students, Professor
Description:	This use case allows users to view the menu of items along with its price.
Trigger:	User clicks the View Menu button.
Preconditions:	1. Respective user must be authenticated. 2. Menu must have items to view.
Postconditions:	1. Menu items are displayed along with prices.
Normal Flow:	1. User opens the application and clicks on the View Menu button. 2. The page containing the menu items along with the price loads up. 3. The administrator alone can add/ remove items from the menu. 4. After making changes the admin clicks the Save to save to

Alternative Flows:	2.1 1. The menu page does not load. 2. The user refreshes the browser.
Exceptions:	
Assumptions:	User has an internet connection to access the application.
Notes and Issues:	

USE CASE 4: Update menu

Use Case ID:	4		
Use Case Name:	Update menu		
Created By:		Last Updated By:	
Date Created:	10/22/2018	Date Last Updated:	10/22/2018

Actors:	Administrator
Description:	This use case allows administrator to update the menu.
Trigger:	Administrator clicks the Update Menu button.
Preconditions:	The administrator is required to add or change items in the menu.

Postconditions:	Items in the menu are updated.
Normal Flow:	1. Administrator clicks the Update Menu button. 2. The page for updating the menu items loads up. 3. The admin makes the required changes and clicks on Save button. 4. The menu is updated.
Alternative Flows:	2.1 1. The update menu items page does not load. 2. The admin refreshes the browser. 3.1 1. The save action does not go through. 2. Menu is not updated. 3. The update menu page is refreshed.
Exceptions:	
Assumptions:	Administrator has an internet connection to access the application.
Notes and Issues:	

USE CASE 5: Approve Payment

Use Case ID:	5		
Use Case Name:	Approve payment		
Created By:		Last Updated By:	
Date Created:	10/22/2018	Date Last Updated:	10/22/2018

Actors:	Accountant

Description:	This use case allows accountants to approve the student purchases
Trigger:	Student makes a personal or project purchase
Preconditions:	Student is required to make a purchase which is then forwarded for approval
Postconditions:	Accountant verifies the purchases and approves it
Normal Flow:	1. Student requires materials from the Innovation Hub 2. Student requests for the material online 3. There are 2 types of purchases possible: Personal and Project 4. After the request is sent, account verifies and approves the purchase
Alternative Flows:	2.1 1. There was no record of the material purchase request 2. Account denies the request
Exceptions:	
Assumptions:	Buyer is an ASU student and needs materials either for a personal project or a course project
Notes and Issues:	

USE CASE 6: Register course project

Use Case ID:	6		
Use Case Name:	Register course project		
Created By:		Last Updated By:	

Date Created:	10/22/2018	Date Last Updated:	10/22/2018

Actors:	Professor
Description:	This use case allows professors to register a course project with the Innovation Hub
Trigger:	Course project requires materials to be purchased and used
Preconditions:	Professor needs a team of students for a course project
Postconditions:	Professor registers the course project and gives the names of the students to Innovation Hub
Normal Flow:	1. Student requires materials from the Innovation Hub 2. Student requests for the material online 3. There are 2 types of purchases possible: Personal and Project 4. After the request is sent, account verifies and approves the purchase
Alternative Flows:	2.1 1. There was no record of the material purchase request 2. Account denies the request
Exceptions:	None
Assumptions:	Buyer is an ASU student and needs materials either for a personal project or a course project
Notes and Issues:	-

USE CASE 7: Make Purchase

Use Case ID:	7

Use Case Name:	Make Purchase		
Created By:		Last Updated By:	
Date Created:	10/24/2018	Date Last Updated:	10/24/2018

Actors:	Student
Description:	This use case describes the process of making payment.
Trigger:	A user requests to make a purchase.
Preconditions:	1. The user has selected the material to purchase. 2. The user has selected the mode of purchase.
Postconditions:	1. The order for material is placed. 2. The payment is sent to the accounting to get reflected in the purchaser's ASU account.
Normal Flow:	1. The user (Student/Professor), submits the form to make purchase. 2. The quantity of the material is updated in the inventory to reflect the purchase. 3. Once the database is updated, it means that the order is placed. 4. The generated cost is forwarded to the accounting to

	approve the payment and bill the user on their ASU account
Alternative Flows:	3. a. The database is not updated due to connectivity issues. b. The order is rolled back.
Exceptions:	-
Assumptions:	The user has provided correct details while making the purchase.
Notes and Issues:	-

USE CASE 8: Validate Purchaser

Use Case ID:	8		
Use Case Name:	Validate Purchaser		
Created By:		Last Updated By:	
Date Created:	10/24/2018	Date Last Updated:	10/24/2018

Actors:	
Description:	This use case describes the process of validation after a

	student makes a purchase in the project purchase mode.
Trigger:	A user requests to make a project purchase.
Preconditions:	1. The student has selected the material to purchase. 2. The student is making a purchase in project purchase mode.
Postconditions:	The student is verified as a member in a course project.
Normal Flow:	1. The details of the student are received from the project purchase form. 2. The system searches for the credentials against the list of projects and displays the list of projects the student is part of. 3. The student is verified and is allowed to select a project from the list.
Alternative Flows:	1. The student is not part of any project.
Exceptions:	-
Assumptions:	-
Notes and Issues:	-

USE CASE 9: Update inventory Dashboard

Use Case ID:	9		
Use Case Name:	Update inventory dashboard		
Created By:		Last Updated By:	
Date Created:	10/24/2018	Date Last Updated:	10/24/2018

Actors:	
Description:	This use case describes the process of updating the inventory based on actions like purchase, menu update and restocking/manual updating.
Trigger:	1. A user make a payment for purchase. 2. The admin updates the menu. 3. The admin restocks/manual updates the inventory.
Preconditions:	1. The user/admin has access to the portal via the internet.
Postconditions:	The dashboard displays latest data from the inventory.
Normal Flow:	1. A change to the inventory has to occur due to processes like making payment, menu update and

	inventory restocking/manual updating. 2. The database is updated. 3. Once refreshed, the dashboard displays updated values.
Alternative Flows:	The database is not updated due to connectivity issues.
Exceptions:	-
Assumptions:	-
Notes and Issues:	-

USE CASE 10: Update Course Project Details

Use Case ID:	10		
Use Case Name:	Update course project details		
Created By:		Last Updated By:	
Date Created:	10/24/2018	Date Last Updated:	10/24/2018

Actors:	Professor
Description:	This use case describes the process where the professor updates the details of the course project team

Trigger:	Professor finalizes the team for the project
Preconditions:	Professor has to register a course project in order to proceed further
Postconditions:	The project team details are updated at the Innovation Hub and they can make purchases accordingly
Normal Flow:	1. Professor registers a project for the course 2. A student team is finalized for the course project 3. Professor updates the project details accordingly 4. This detail is then forwarded to the Innovation Hub
Alternative Flows:	1.1 1. Professor registers a project for the course 2. Due to some reason, professor has to cancel the project 1.2 1. Professor registers a project for the course 2. Team is finalized for the project 3. Before the start of the project, team backs out of the project
Exceptions:	None

Assumptions:	Project has been registered for the course
Notes and Issues:	-

USE CASE 11: View Course Projects

Use Case ID:	11		
Use Case Name:	View course projects		
Created By:		Last Updated By:	
Date Created:	10/24/2018	Date Last Updated:	10/24/2018

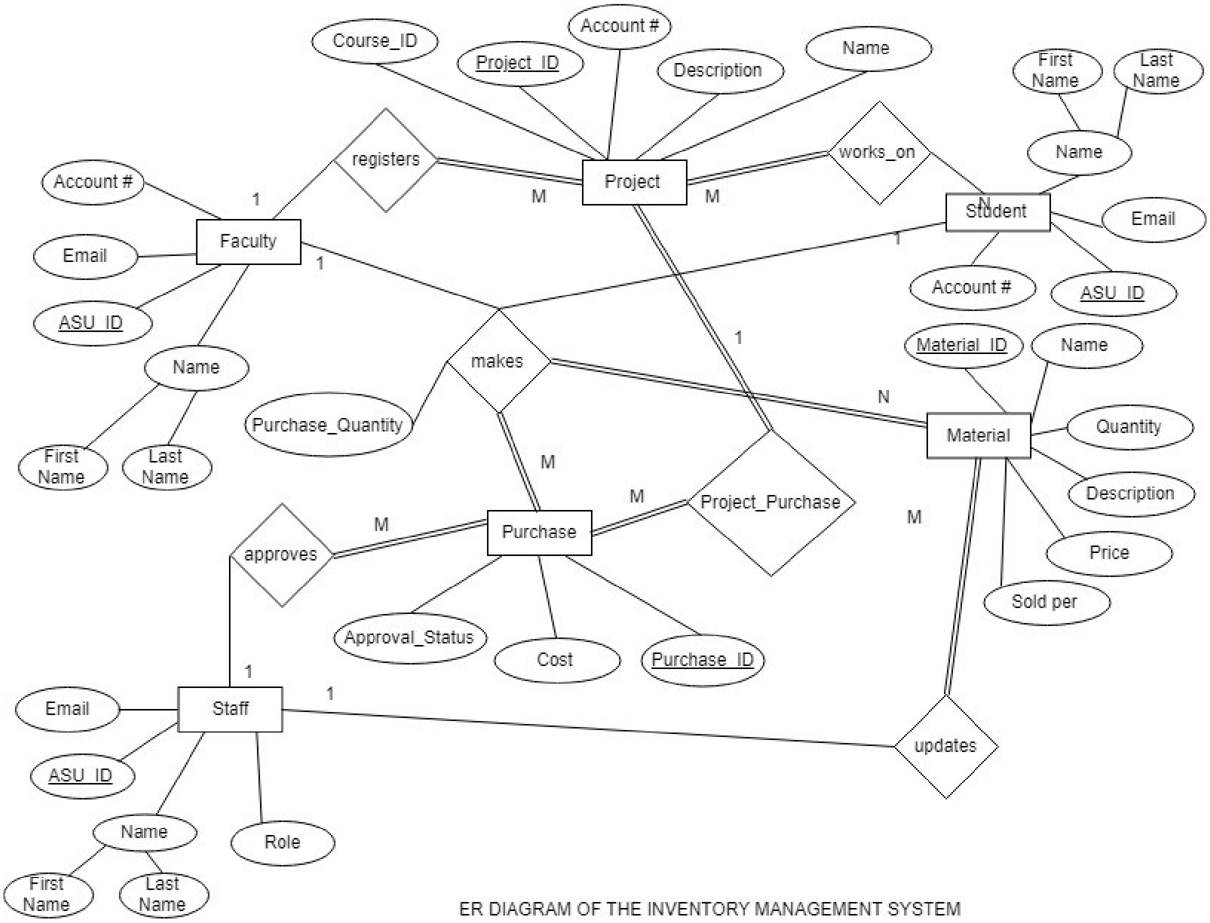
Actors:	Professor
Description:	This use case describes the process of viewing the course projects set by the professor for a particular course
Trigger:	There are a lot of groups having different course projects
Preconditions:	Course should have projects
Postconditions:	List of projects and team details can be viewed by the professor

Normal Flow:	1. Professor registers projects for the course 2. Teams are formed and course project details are updated 3. Professor is then able to view the details
Alternative Flows:	Professor is unable to view the details
Exceptions:	-
Assumptions:	Different teams have each been alloted projects
Notes and Issues:	-

SECTION 3: STRUCTURING DATA REQUIREMENTS

Conceptual Entity-Relationship Schema Diagram

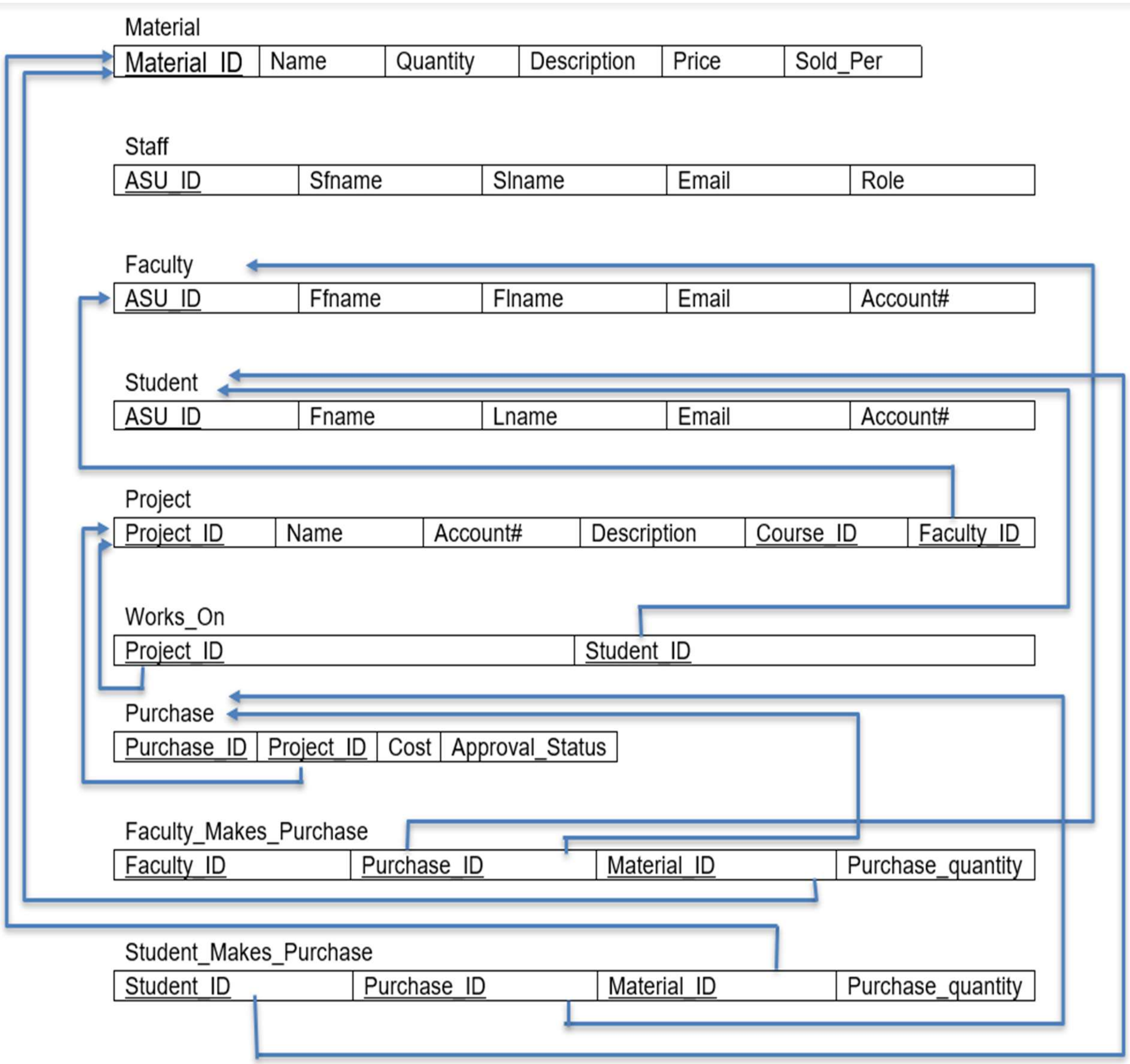
Based on the requirements gathered in previous phases, we developed a ER diagram. The entities, relationships and their attributes are all depicted in standard notation used in ER Diagram.



Conceptual ER Diagram

ER-to-Relational Mapping

After designing the ER diagram for our system, it was transformed into a Relational model which can then be implemented by RDBMS.



Material

Material_ID	Name	Quantity	Description	Price	Sold_Per

Staff

ASU_ID	Sfname	Slname	Email	Role

Faculty

ASU_ID	Ffname	Flname	Email	Account#

Student

ASU_ID	Fname	Lname	Email	Account#

Project

Project_ID	Name	Account#	Description	Course_ID	Faculty_ID

Works_On

Project_ID	Student_ID

Purchase

Purchase_ID	Project_ID	Cost	Approval_Status

Faculty_Makes_Purchase

Faculty_ID	Purchase_ID	Material_ID	Purchase_quantity

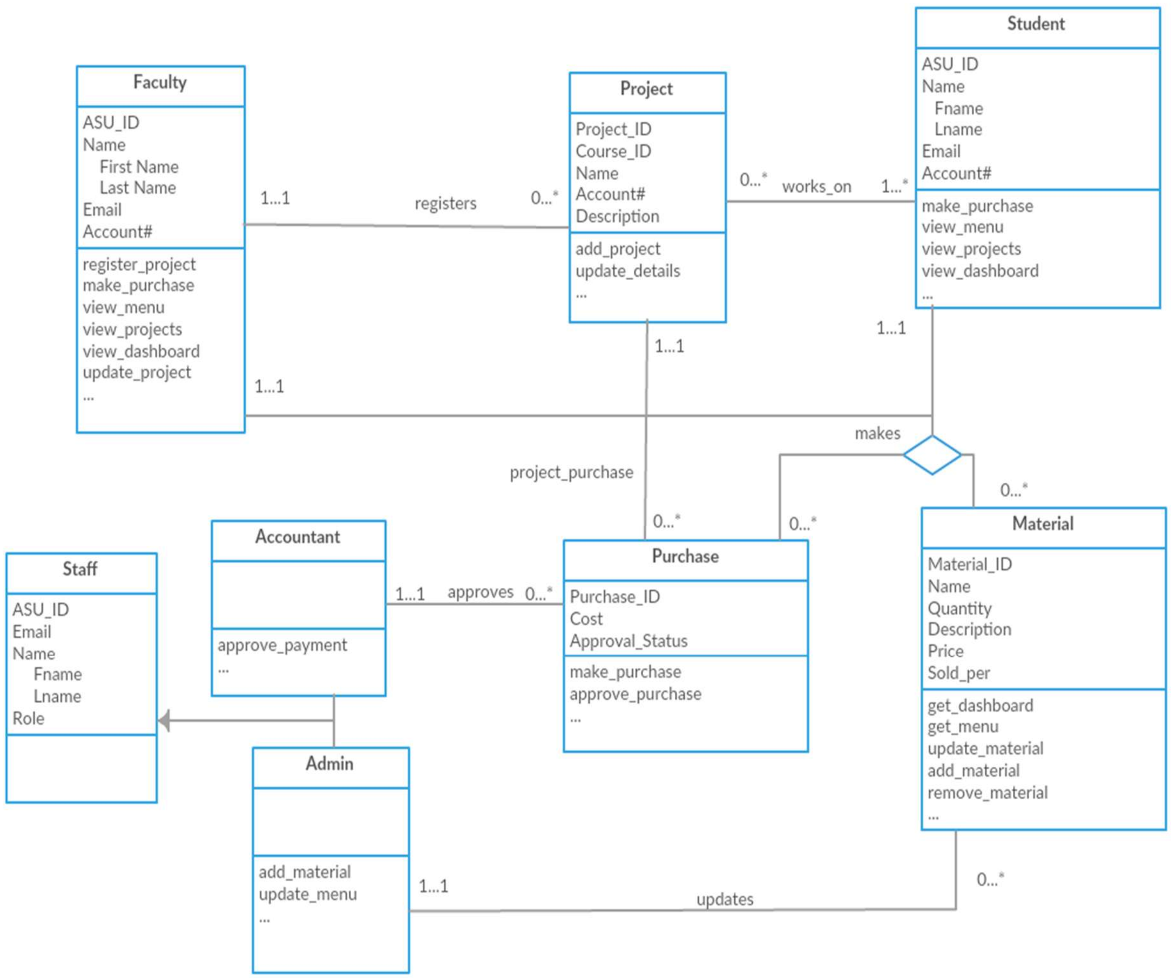
Student_Makes_Purchase

Student_ID	Purchase_ID	Material_ID	Purchase_quantity

Relational Schema

UML Class Diagram

By using the ER Diagram as reference, the Class Diagram was developed. Class diagrams clearly map out the structure of a particular system by modelling its classes, attributes and relationships between objects.



Class Diagram

Implementation of database

Physical implementation of the database has been performed using MySQL. The SQL script for the same has been provided in the Appendix at the end of this report.

SECTION 4: USER INTERFACE

After implementing the physical database, a prototype of the user interface based on use case scenarios was developed. The UI was developed in compliance to the 6 Design Principles for UI Design and incorporates the 3 fundamental parts associated with any user interface viz. Navigation mechanism, Input mechanism, and Output mechanism.

The 6 Design Principles are:

1. Layout
2. Content Awareness
3. Aesthetic
4. User Experience
5. Consistency
6. Minimal User Efforts

Use Scenarios

The following use scenarios were incorporated into the User Interface:

Use case name: View and update inventory
Actor: Administrator
Normal Course:

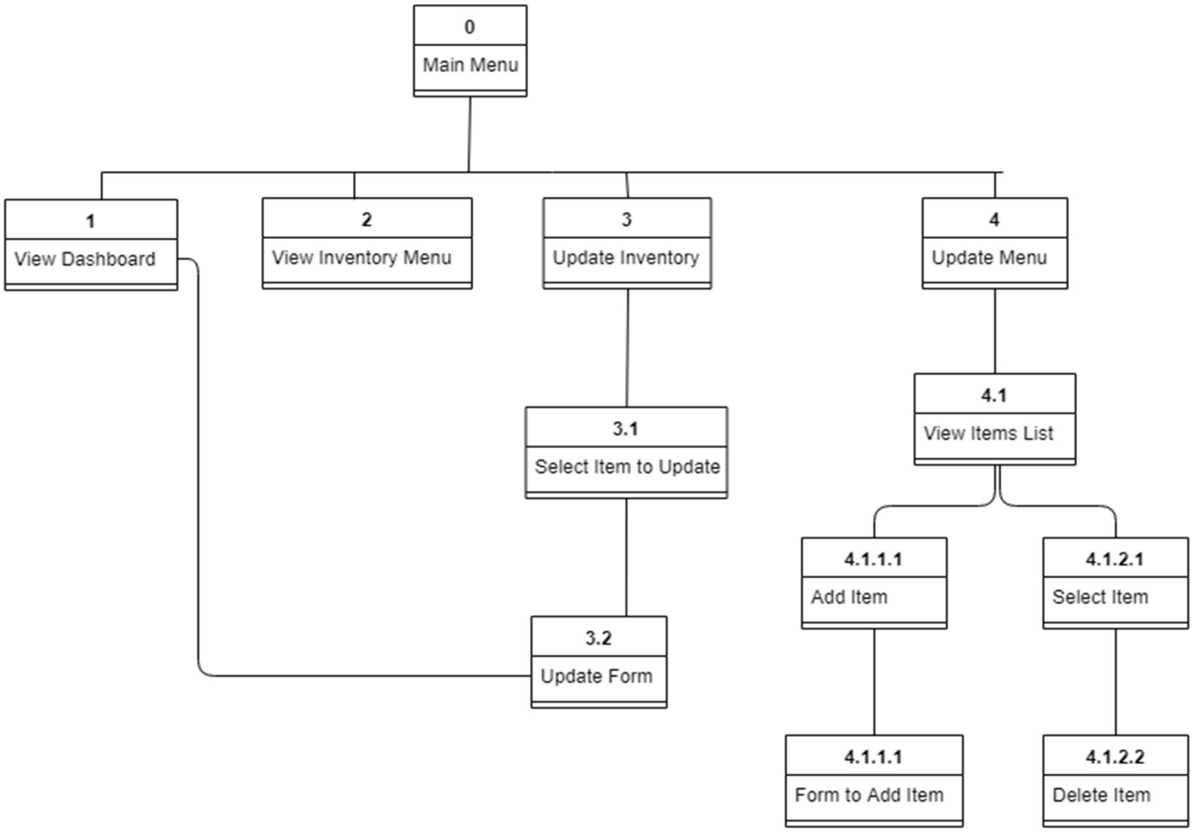
1. System displays the home page, by default displays the dashboard.
2. The admin can view the dashboard in the home page.
3. The admin clicks update inventory to navigate to the inventory update window.
4. Admin searches the item to update in the search box.
5. Admin enters the quantity against the item that requires update.
6. Admin submits the updates.

Use case name: View and update menu
Actor: Administrator
Normal Course:

1. System displays the home page.
2. Admin clicks Inventory Menu to view the menu items.
3. Admin clicks Update Menu to navigate to the update menu window.
4. Admin clicks Add Item to add new item to the menu.
5. Admin inputs the details and submits.
6. From the update menu window admin selects the item to delete.
7. Admin performs delete.

Interface Structure Design

Based on the use scenarios, we developed the Interface Structure Design:



Interface Structure Design

It is evident that the Interface Structure Design implemented is consistent with the Use Scenarios and with the overall UI of the system.

Interface Standards

Interface standards that we have maintained across the system are as follows:

Interface Metaphor:

Innovation Hub Inventory Management

Interface Objects:

Students, Faculty, Administrators, Inventory, Materials (Item)

Interface Actions:

The admin can perform the following actions

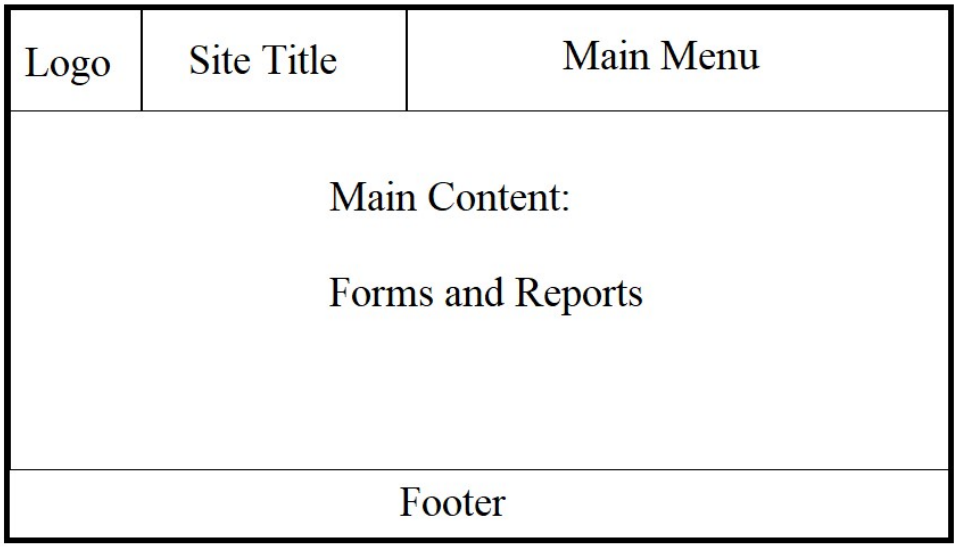
- View inventory
- Update inventory
- View menu
- Update menu

Interface Icons:

The application uses a metaphor logo at the to left of the interface

Interface Templates:

The application follows uniform placement of contents as designed in the wireframe. The color theme used in the interface is consistent throughout.

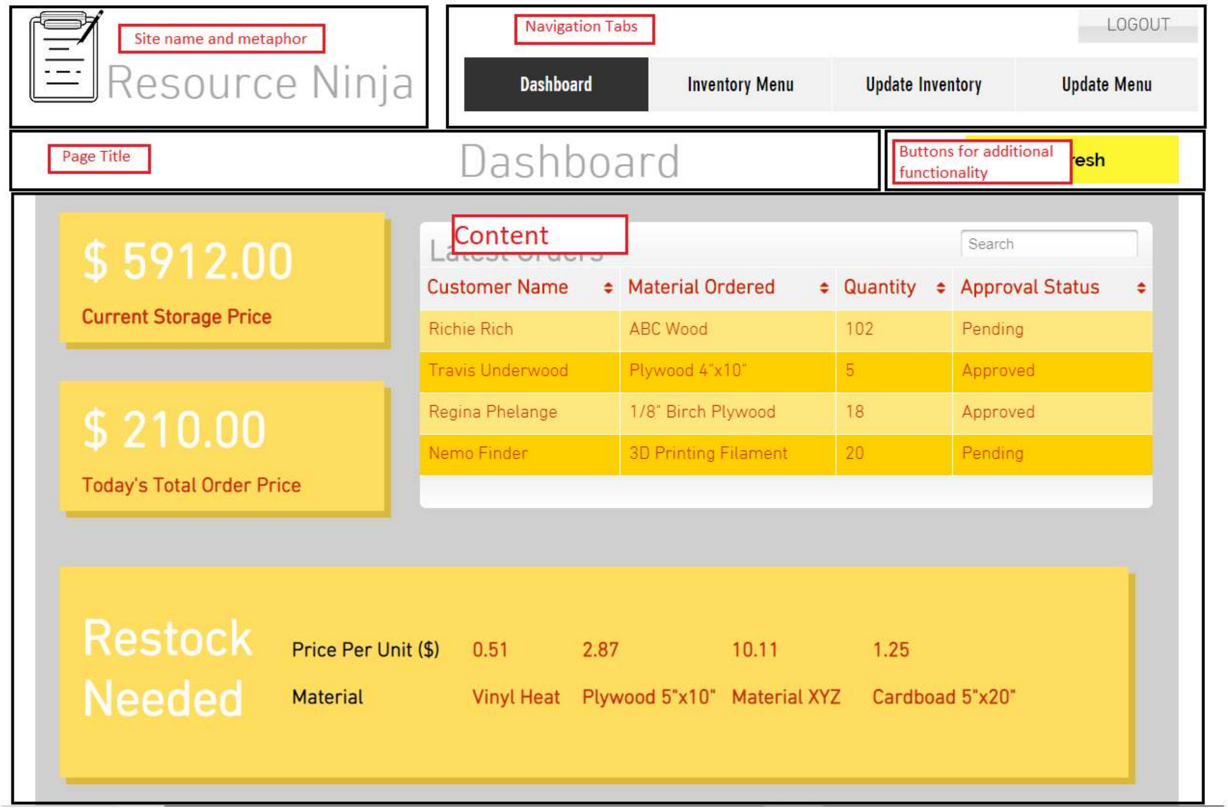


Interface Design Prototypes

Wireframe Diagram:

The Wireframe diagram of the UI is shown below:

Inventory Management Home Page:



Dashboard

$ 5912.00

Current Storage Price

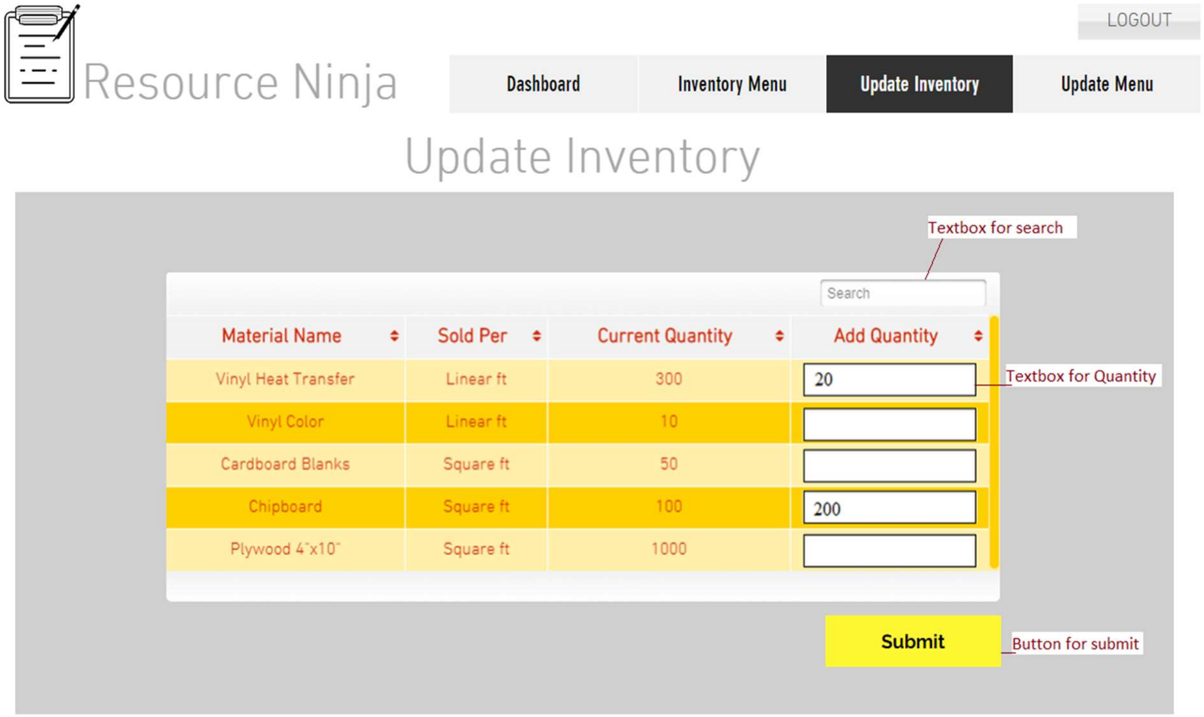
$ 210.00

Today's Total Order Price

Customer Name	Material Ordered	Quantity	Approval Status
Richie Rich	ABC Wood	102	Pending
Travis Underwood	Plywood 4"x10"	5	Approved
Regina Phelange	1/8" Birch Plywood	18	Approved
Nemo Finder	3D Printing Filament	20	Pending

Restock Needed

Price Per Unit ($)	0.51	2.87	10.11	1.25
Material	Vinyl Heat	Plywood 5"x10"	Material XYZ	Cardboad 5"x20"



Update Inventory

Material Name	Sold Per	Current Quantity	Add Quantity
Vinyl Heat Transfer	Linear ft	300	20
Vinyl Color	Linear ft	10	
Cardboard Blanks	Square ft	50	
Chipboard	Square ft	100	200
Plywood 4"x10"	Square ft	1000	

Textbox for search

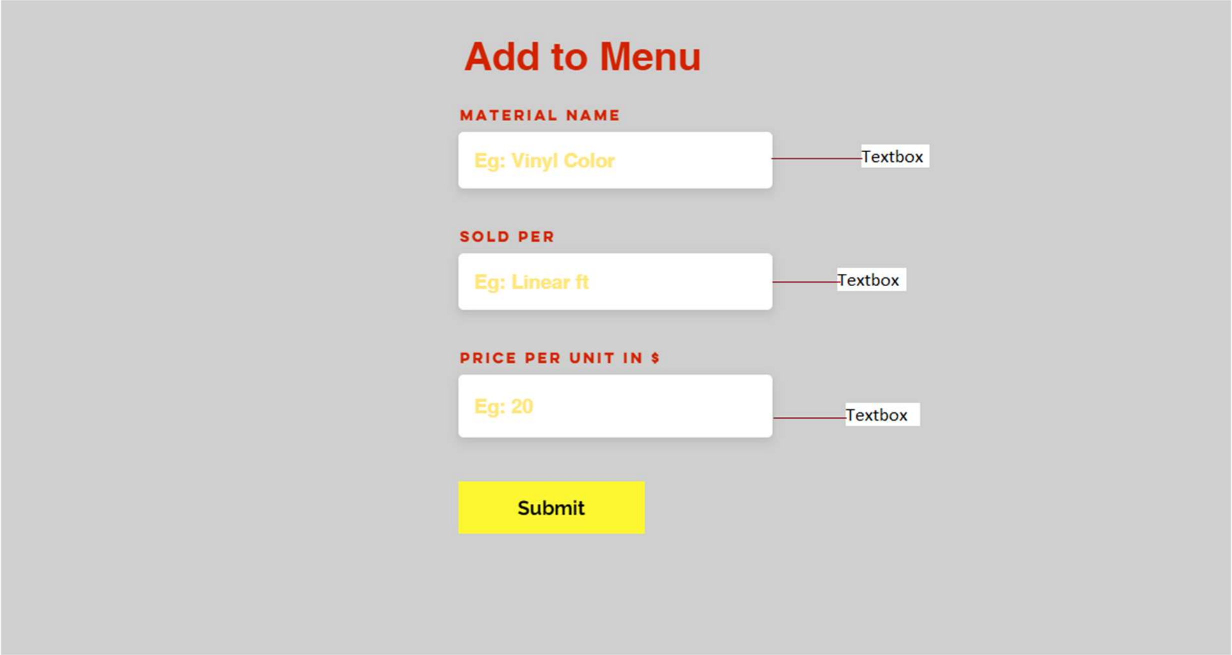
Textbox for Quantity

Button for submit

Updating Inventory:

Updating Menu:

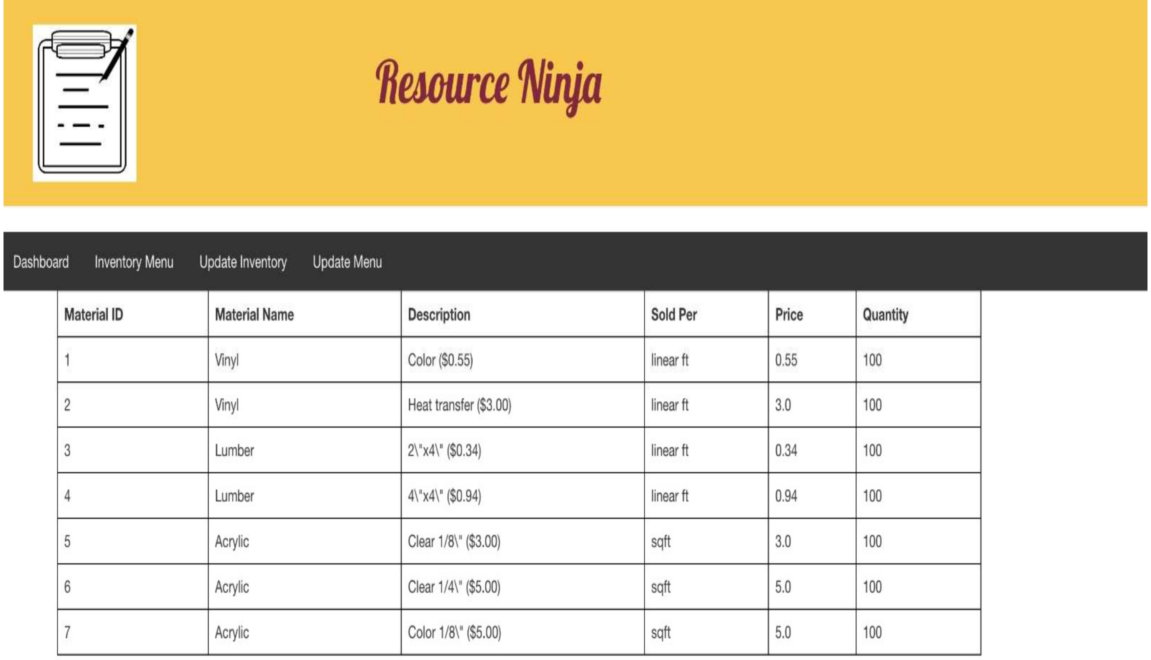
Add New Item to Menu



HTML Prototypes

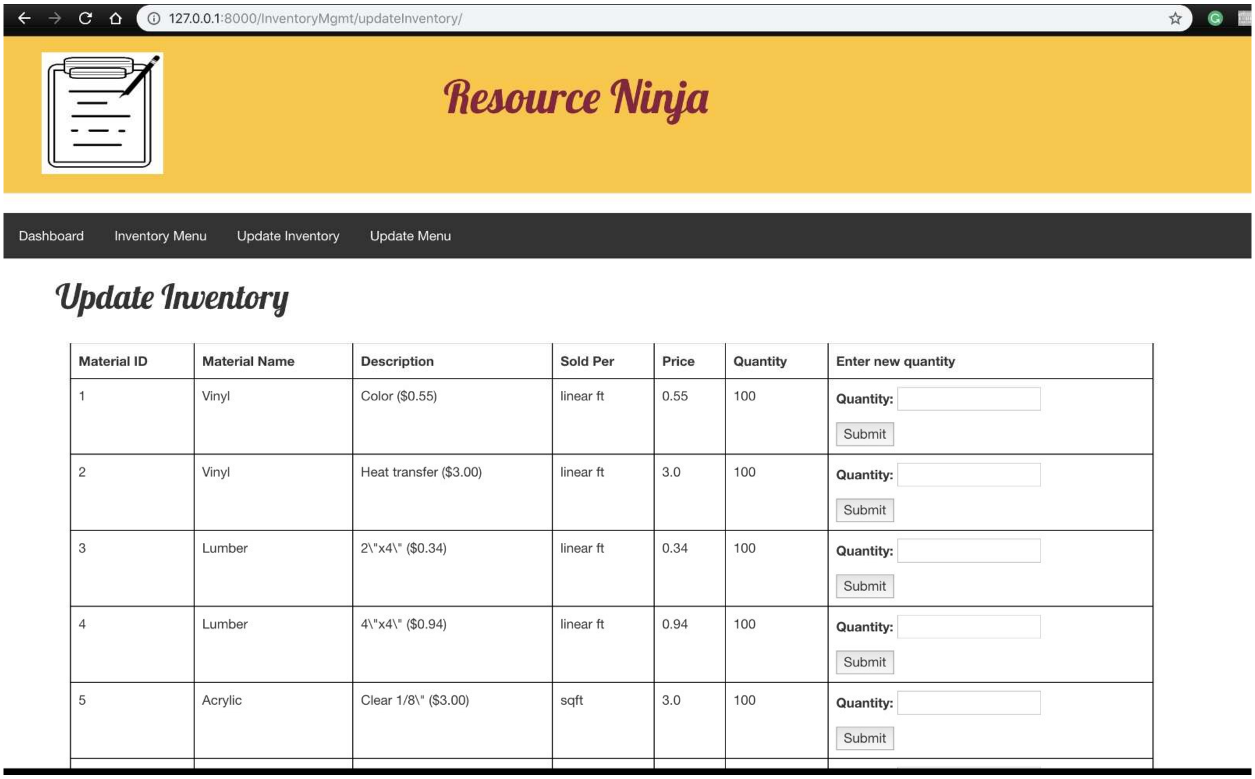
Based on the Wireframe diagram, we developed HTML prototypes for our system. The screenshots are shown below:

Dashboard



Material ID	Material Name	Description	Sold Per	Price	Quantity
1	Vinyl	Color ($0.55)	linear ft	0.55	100
2	Vinyl	Heat transfer ($3.00)	linear ft	3.0	100
3	Lumber	2\"x4\" ($0.34)	linear ft	0.34	100
4	Lumber	4\"x4\" ($0.94)	linear ft	0.94	100
5	Acrylic	Clear 1/8\" ($3.00)	sqft	3.0	100
6	Acrylic	Clear 1/4\" ($5.00)	sqft	5.0	100
7	Acrylic	Color 1/8\" ($5.00)	sqft	5.0	100

Update Inventory

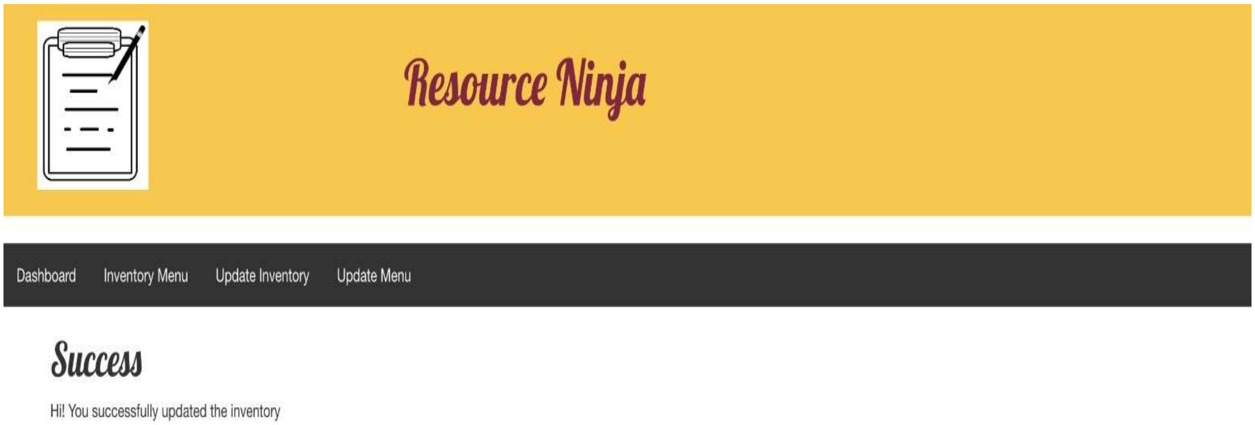


Resource Ninja

Update Inventory

Material ID	Material Name	Description	Sold Per	Price	Quantity	Enter new quantity
1	Vinyl	Color ($0.55)	linear ft	0.55	100	**Quantity:** Submit
2	Vinyl	Heat transfer ($3.00)	linear ft	3.0	100	**Quantity:** Submit
3	Lumber	2\"x4\" ($0.34)	linear ft	0.34	100	**Quantity:** Submit
4	Lumber	4\"x4\" ($0.94)	linear ft	0.94	100	**Quantity:** Submit
5	Acrylic	Clear 1/8\" ($3.00)	sqft	3.0	100	**Quantity:** Submit

Successful Update



Resource Ninja

Success

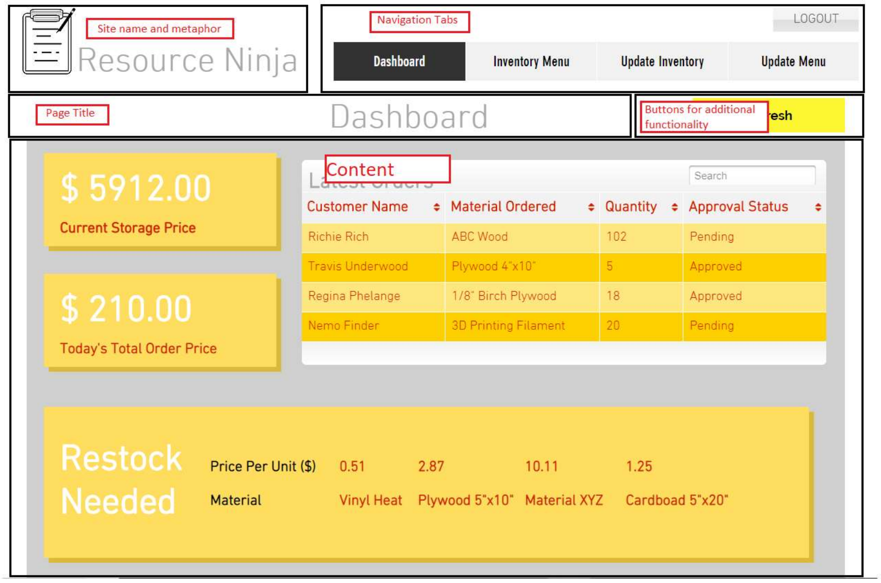
Hi! You successfully updated the inventory

Interface Evaluation

Design of Interface and Adherence to Design Principles:

The User Interface adheres to the 6 design principles. Proof of the same has been given below:

1. **Layout**



$ 5912.00
Current Storage Price

$ 210.00
Today's Total Order Price

Customer Name	Material Ordered	Quantity	Approval Status
Richie Rich	ABC Wood	102	Pending
Travis Underwood	Plywood 4"x10"	5	Approved
Regina Phelange	1/8" Birch Plywood	18	Approved
Nemo Finder	3D Printing Filament	20	Pending

Restock Needed

Price Per Unit ($)	0.51	2.87	10.11	1.25
Material	Vinyl Heat	Plywood 5"x10"	Material XYZ	Cardboad 5"x20"

As evident from the screenshot above, the layout of the user interface clearly contains 3 distinct areas: the top area which is used for navigation, the left side used for local navigation, the middle area where the main content of the webpage is displayed, and the bottom area. This layout structure has been maintained consistently across all the pages of the website.

2. **Content Awareness**

All the webpages, which serve as a medium for users to communicate with the underlying inventory, contain distinct and appropriate titles. All the input fields in the forms are clearly labelled and users can easily identify the use of an input field.

3. **Aesthetics**

   The User Interface has been designed in a simple and minimalistic manner to make it aesthetically pleasing to the user. Also, we have made sure that none of the features have been compromised in achieving the same. Proper font sizes and colors have been used at the correct places.

4. **User Experience**

   The website is easy to understand and use for different sets of users.

5. **Consistency**

   The base layout for all the pages is the same and consistent color layout is used across the webpage.

6. **Minimizing User Effort**

   Our User Interface follows the 3-click rule wherein users are able to reach their desired page by using 3 mouse clicks at max.

SECTION 5: ARCHITECTURE

Non-Functional Requirements

2.4. Operational

2.4.1.1. System should be compatible with majority of web browsers

2.5. Performance

2.5.1.1. System should be able to handle multiple order requests at the same time

2.6. Security

2.6.1.1. System requires login for authentication

2.6.1.2. Providing access according to user roles

2.4. Cultural and Political

2.4.1. System must be in compliance with local laws and regulations pertaining to information privacy and other content.

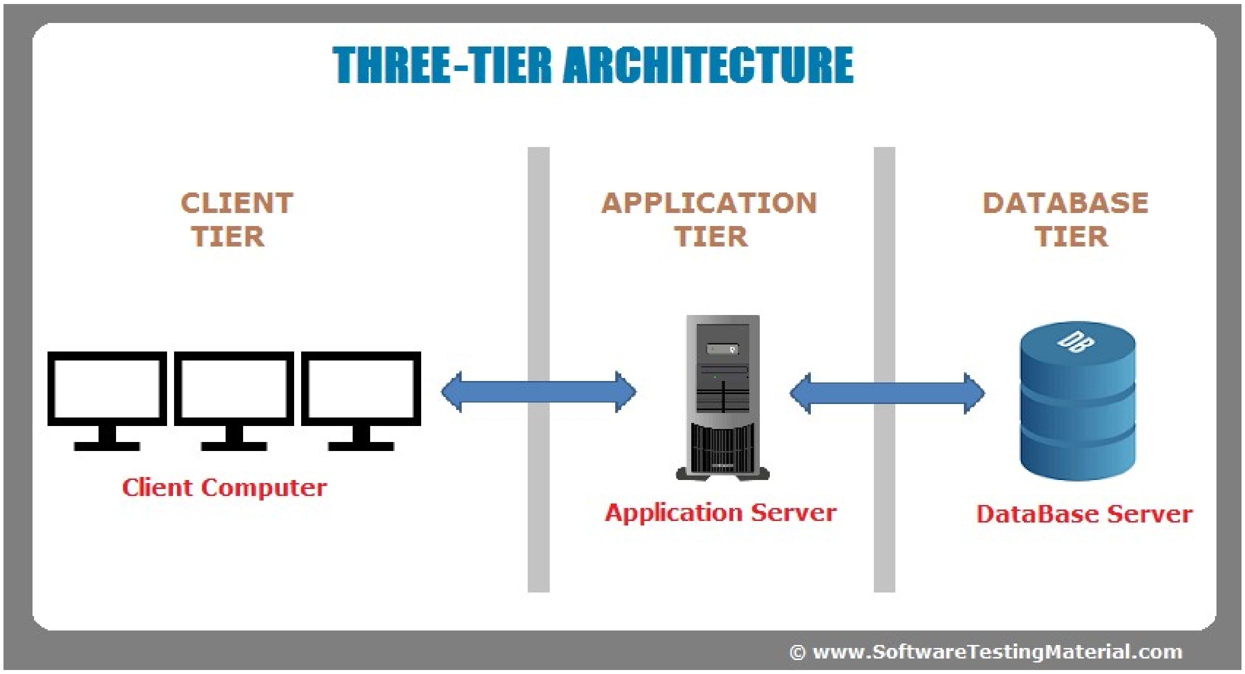
Architecture Used

The architecture used for our system is a Three-Tier architecture. A three-tier architecture is a client-server architecture in which process logic, data storage, user interface etc. are developed and maintained as independent modules on separate platforms. The user interface is implemented on a desktop PC and uses a standard GUI with different modules running on the application server. The three tiers are:

1. Presentation Tier

2. Application Tier

3. Data Tier



Architectural Diagram

SECTION 6: IMPLEMENTATION

Coding and Documentation

The Model-View-Template (MVT) framework is used to design the application. There are three logical components: model, view and template, each of which handles specific portion of the application.

Model: This represents a single, definitive source of information about data. It contains the essential fields and behaviors of the data that need to be stored.

View: This is a function that takes a Web request and returns a Web response. This response can be the HTML contents of a Web page, or a redirect, or a 404 error, or an XML document, or an image. The view itself contains logic that is necessary to return that response.

Template: Templates contains the static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted and templates interact with the Views to render the final output.

User Guide

The code was pushed to the github directory: https://github.com/shreya2592/RNinja Please follow the below steps to download and run the project

1. Clone the code from the github to local machine using the following command in the terminal: git clone https://github.com/shreya2592/RNinja.git
2. Install python3 in your local machine (If you don't have) from: https://www.python.org/downloads/
3. Install Django from terminal using: pip3 install Django
4. You can install the database MYSQL as: https://dev.mysql.com/downloads/connector/python/
5. After all the installation go inside the RNinja folder and run the following command: python3 manage.py runserver
6. Go to the browser and put in the following URL to check the web application: http://127.0.0.1:8000

Modules

Below is the description of the different modules implemented:

1. Login: The users login to the application from this page.
2. View Inventory: The admin can view the list of inventory along with other details as well.
3. Update Inventory: The admin can make changes to the inventory here.
4. View Menu: The admin and purchasers can view the list of items available to purchase.
5. Edit Menu: The admin can make changes to the menu here.

Hardware Requirements

Any laptop or computer with JavaScript enabled web browser can be used to run this application

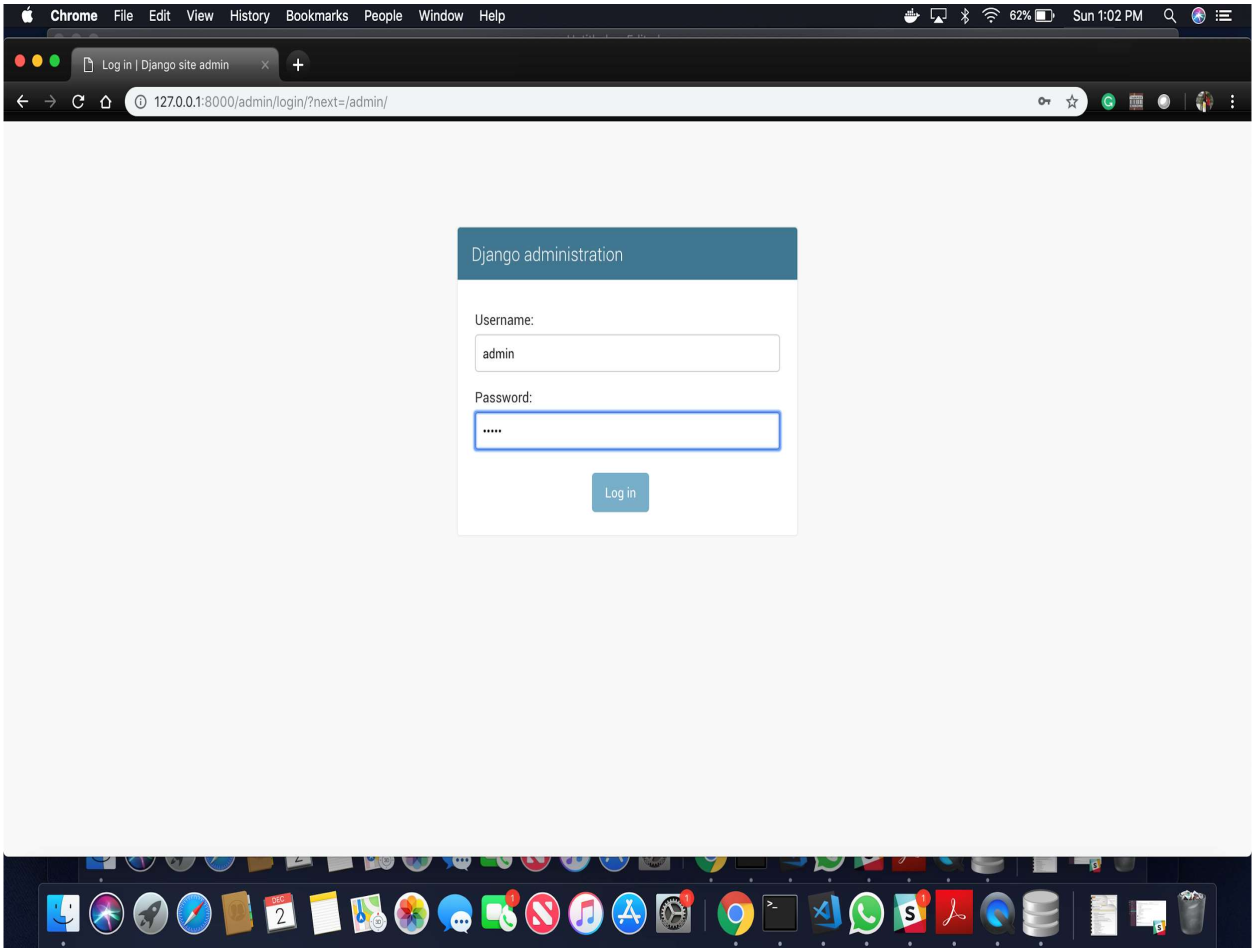
Software Requirements

The software required to run this application are:

- BackEnd Framework & Routing Requests: Django
- Database: MySQL
- FrontEnd: HTML/CSS

Output Screenshots

Login

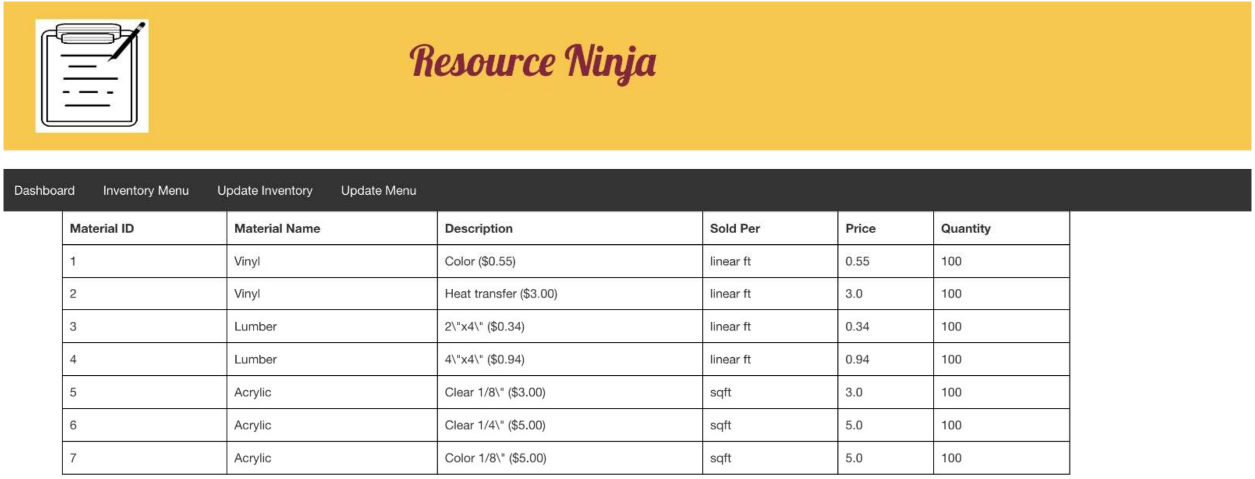

Chrome File Edit View History Bookmarks People Window Help
62% Sun 1:02 PM
Log in | Django site admin
127.0.0.1:8000/admin/login/?next=/admin/
Log in

Django administration

Username: admin

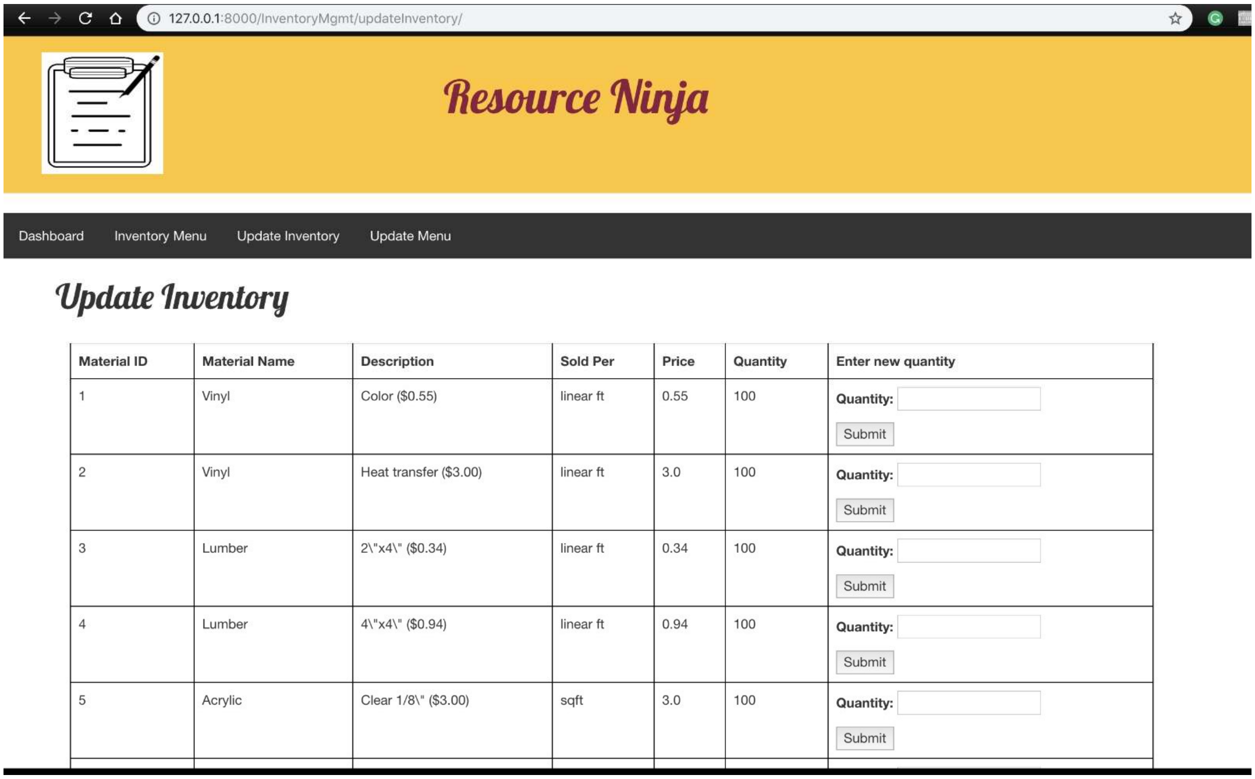
Password: •••••

Inventory Dashboard



Material ID	Material Name	Description	Sold Per	Price	Quantity
1	Vinyl	Color ($0.55)	linear ft	0.55	100
2	Vinyl	Heat transfer ($3.00)	linear ft	3.0	100
3	Lumber	2\"x4\" ($0.34)	linear ft	0.34	100
4	Lumber	4\"x4\" ($0.94)	linear ft	0.94	100
5	Acrylic	Clear 1/8\" ($3.00)	sqft	3.0	100
6	Acrylic	Clear 1/4\" ($5.00)	sqft	5.0	100
7	Acrylic	Color 1/8\" ($5.00)	sqft	5.0	100

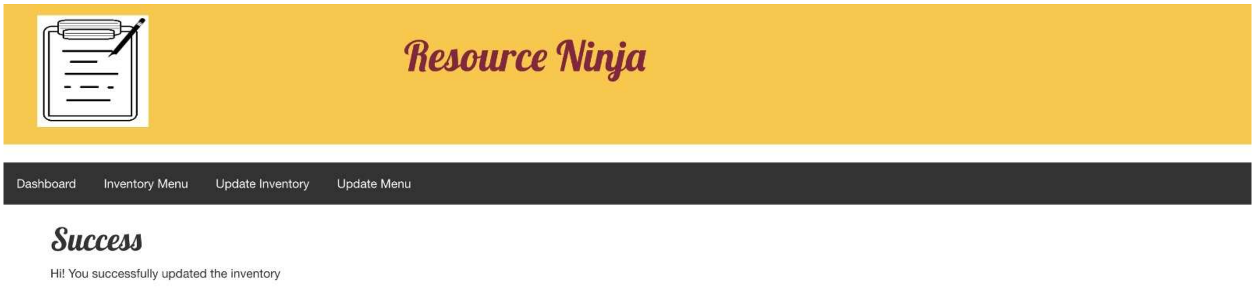
Update Inventory



Update Inventory

Material ID	Material Name	Description	Sold Per	Price	Quantity	Enter new quantity
1	Vinyl	Color ($0.55)	linear ft	0.55	100	Quantity: Submit
2	Vinyl	Heat transfer ($3.00)	linear ft	3.0	100	Quantity: Submit
3	Lumber	2\"x4\" ($0.34)	linear ft	0.34	100	Quantity: Submit
4	Lumber	4\"x4\" ($0.94)	linear ft	0.94	100	Quantity: Submit
5	Acrylic	Clear 1/8\" ($3.00)	sqft	3.0	100	Quantity: Submit

Success Message



Success

Hi! You successfully updated the inventory

CONCLUSION

Accomplishments

We are successfully able to help Innovation Hub's Inventory Management in transitioning from a totally manual resource management system to a system with automated workflows. We have a fully developed database that can be used for the complete implementation.

Future Scope

Implementing all the identified requirements for the Inventory Management system, such as components for students, faculty and accountant. Additional features like Prepayment and Reservation of material. Also, future scope involves development of the Scheduling and Quoting system.

Recommendations

The application is designed to work on a web browser. We could develop a mobile app which would improve the accessibility of the application.

Challenges overcome

We are building the system from ground up and therefore, we had to understand the system from scratch and come up with all the non-functional and functional requirements. It was difficult to find a point of reference. Discovering the workflows and designing automated solutions were the major objectives of the project which were pretty challenging. We had difficulties in understanding the independent and dependent workflows as our system incorporated multiple actors and actions. Coming up with the relationships and cardinalities. Designing the wireframe in a way that it adhered to the design principles.

REFERENCE

- Modern System Analysis and Design- 8th edition - Valacich, George
- Lecture notes - Requirements
- Lecture notes - Structuring Requirements
- Lecture notes - Structuring Data Requirements
- Lecture notes - Class Diagram
- Lecture notes - ER to Relational
- Lecture notes - User Interface

APPENDIX

A. Minutes of the Meeting

a. Sean Dengler - 10th October 2018

Interviewee(s): Sean Dengler
Interviewer(s):

<table>
<tr><td colspan="3">Shreya Srivastava, Aditya Kalyanaraman, Venkata Siva Abhishek Munukutla, Mohan Suraj Lakshmi Narayana, Ramya Chandrasekhar</td></tr>
<tr><td colspan="2">Purpose of the interview:

To learn about the current system in place and look for areas of improvement.</td><td>Date:

10th October, 2018</td></tr>
<tr><td>Questions</td><td>Answers</td><td>Observations/Notes</td></tr>
<tr><td>How does the current system work?</td><td>The current system is completely manual. Different components are handled independently.</td><td>Sean manages the system with the help of various online tools, spreadsheets and human intervention.</td></tr>
<tr><td>Apart from you, who else in involved in managing the current system?</td><td>Sean is the only point of contact. If Sean leaves, the system falls apart.</td><td>Although there are other employees in the innovation hub, Sean is the only one who knows how the system is working.</td></tr>
<tr><td>What tools do you use in the current system?</td><td>Google Forms, Google Docs (Excel) and Schedulerthing.com</td><td>There are some parts like the materials he is purchasing is not documented.</td></tr>
<tr><td>Do you think the existing scheduler system can be improved?</td><td>Yes, Sean believes so.</td><td>An external vendor application is being used. It would be good to have our own system that is in house.</td></tr>
<tr><td>What changes would you like to see in the current scheduler system?</td><td>The scheduler system is a 3rd party system. It would be great to have a system built by ASU to reduce security risks.

The system also doesn’t have a login system for students/users. A user is allowed to book 4 slots of half hour each at once. Due to lack of login, a user can book multiple times and block a machine for the whole day.</td><td>There were a number of changes in logic and functionality. As we move forward, we will have more questions.</td></tr>
</table>

	The system doesn't allow the scheduler to cancel the booking. Sean has to manually go in as the administrator and cancel their schedule. It takes time to enquire and confirm the schedules via email. This creates plenty of delay in the process and the machines are blocked out from being used.	
Do you want to integrate the current scheduler with an inventory management system?	No, Sean doesn't want integration. The inventory should remain independent as long as the scheduler is third party.	Our initial understanding of the dependency of the system is different from what we discussed today. The 2 components are independent.
What do you think is the most important for the lab - a scheduler or an inventory management system?	According to Sean, the inventory management system will solve the complexities of manual work. A point of sale would really help the lab verify and track the purchases. It would also help in restocking the inventory.	ASU s money is invested in the innovation hub. Proper tracking of the funds is important.
How does the current inventory management system work?	There is no database that tracks the quantity of materials and purchases. Sean uses Google forms for purchase requests and excel sheets to record the purchases. Sean visually tracks the inventory, i.e., the decision of restocking is autonomous.	We were able to brainstorm and come up with good ideas. Future sessions will be very valuable.
What do you think should the inventory database contain?	The inventory should keep a track of material in the lab.	Keep track of the materials bought and sold.

How do you purchase the material for the lab? (restock)	Sean purchases the material based on personal estimation (visual inspection) or if there is an order. Then, Sean presents the bill to ASU to get a reimbursement.	It would be valuable to create a dashboard to show what materials is sold out.
How often do you purchase the material for the lab?	Sean purchases based on the order or if the lab is running out of stock (during visual inspection).	A more sophisticated method to find out of stock items should be designed and developed.
Do you update the price of the material in the inventory according to the market?	Sean says that the price should be updated but he doesn’t do it frequently. Sean makes changes only if the prices have a huge surge.	More discussion needs to go into this.
How do students and professors make purchases from the lab?	The users of the lab put in a request via Google forms. They can request to purchase either personally or as a team member from a class project. Once the request is approved by Sean, an employee from the accounting department executes the transaction.	There is a lot of human intervention that leads to human errors.
The current system looks like it is prone to human-errors due to its complexity. Don’t you think so?	Yes, a user may enter incorrect or fake information while placing the order. This could create delays and confusion.	Currently, assuming that all the entries are legitimate. Making it part of ASU’s single sign on, dual authentication can be a good idea.
What are the problems with purchase based on class project?	Firstly, Sean has to verify the genuineness of the class project. Then, he must get the list of students from the professor via email exchange. Once the list is fed to the google forms,	There should be a way to automatically authenticate the students that are a part of the project.

	if an order is placed by a student in the group, Sean has to verify from the given list and confirm the order. This process is highly prone to human-errors.	
How do you verify if the buyer is in the professor's class?	As of now, the verification is done manually. Sean must exchange emails with the professors and students to verify.	This is a complex use case to incorporate into the system.
What are the problems with personal purchase?	Personal purchases can sometimes be fake or can contain incorrect information. Confirming the purchase is difficult without verification.	ASU's authentication should be involved here similar to project purchase.
How do you verify the genuineness of the personal purchase?	There is no definite way for verification as of now. Sean has to verify the details of the buyer. If the details are incorrect, Sean has to send the buyer email and get a confirmation from them.	Same as above
Isn't the current system labor intensive?	Yes, it is.	Our system is definitely going to help them. But a lot of thought has to go into implementing it.
Do you think the accounting department needs an approval system to ease the process for both the parties?	Sean thinks so. The accounting department has to confirm the purchases in various excel sheets. They have to use functions in the excel sheets to verify the price and execute the sale. Having various excel sheets makes their job more complex.	We got a lead to connect with Debra, who takes care of the accounting.
Do you think automating	Yes, it will. The system	We will be having sessions

processes can improve the lab's productivity?	will reduce complexity and manual work. Moreover, if it has a login for everyone, the system will be secure, and Sean and the accounting department could avoid verification during the process of purchasing by users.	to brainstorm on the scope and design of the project.

b. Derek Robinson - 10th November 2018

Interviewee(s): Derek Robinson	
Interviewer(s): Shreya Srivastava, Aditya Kalyanaraman, Venkata Siva Abhishek Munukutla, Mohan Suraj Lakshmi Narayana, Ramya Chandrasekhar	
Purpose of the interview: To learn about the current system in place from an end user's perspective.	Date: 10th November, 2018

Questions	**Answers**	**Observations/Notes**
How do you maintain the inventory?	Derek get the requirement by ear and brings in the supply needed .	Similar to what Sean said.
How much time does procurement take?	It usually takes about a week to procure unavailable material.	Similar to what Sean said.

Is there a software that could ease the process?	No. It would be really great if there was one, especially if the tool could streamline and automate the whole process.	Similar to what Sean said.
Would you like the system to reserve material until a date/time?	Derek agrees. It will free up the material for use if not reserved anymore. This saves cost and material.	This could help the lab to put the material to judicious use.
Is there a security concern related to the material?	Not security, but its more about how customers sometimes use material available and they do not report it.	The use is sometimes not recorded.
What else would you like the system to have?	It would be nice to have prepayment for material bought that is not available.	This can be an additional feature.

B. SQL Script for the Database

```
CREATE DATABASE IF NOT EXISTS innovation_hub_inventory;
USE innovation_hub_inventory;

CREATE TABLE IF NOT EXISTS material (
    material_id BIGINT NOT NULL AUTO_INCREMENT UNIQUE,
    material_name VARCHAR(150) NOT NULL,
    description VARCHAR(500),
    sold_per VARCHAR(100),
    price FLOAT NOT NULL,
    quantity FLOAT,
    PRIMARY KEY (material_id)
);


CREATE TABLE IF NOT EXISTS staff(
   asu_id BIGINT NOT NULL AUTO_INCREMENT UNIQUE,
   first_name VARCHAR(150) NOT NULL,
   last_name VARCHAR(150) NOT NULL,
   email VARCHAR(200) NOT NULL UNIQUE,
   staff_password VARCHAR(20) NOT NULL,
   role VARCHAR(100),
   PRIMARY KEY ( asu_id )
);


CREATE TABLE IF NOT EXISTS faculty(
   asu_id BIGINT NOT NULL AUTO_INCREMENT UNIQUE,
   first_name VARCHAR(150) NOT NULL,
   last_name VARCHAR(150) NOT NULL,
   email VARCHAR(200) NOT NULL UNIQUE,
   faculty_password VARCHAR(20) NOT NULL,
   account_number BIGINT NOT NULL UNIQUE,
   PRIMARY KEY ( asu_id )
```

```
);


CREATE TABLE IF NOT EXISTS student(
   asu_id BIGINT NOT NULL AUTO_INCREMENT UNIQUE,
   first_name VARCHAR(150) NOT NULL,
   last_name VARCHAR(150) NOT NULL,
   email VARCHAR(200) NOT NULL UNIQUE,
   student_password VARCHAR(20) NOT NULL,
   account_number BIGINT NOT NULL UNIQUE,
   PRIMARY KEY ( asu_id )
);

CREATE TABLE IF NOT EXISTS project(
   project_id BIGINT NOT NULL AUTO_INCREMENT UNIQUE,
   project_name VARCHAR(150) NOT NULL,
   description VARCHAR(1000) NOT NULL,
   course_id VARCHAR(10),
   faculty_id BIGINT NOT NULL,
   account_number BIGINT NOT NULL,
   PRIMARY KEY ( project_id ),
   FOREIGN KEY (faculty_id)
        REFERENCES faculty(asu_id)
);


CREATE TABLE IF NOT EXISTS purchase(
   purchase_id BIGINT NOT NULL AUTO_INCREMENT UNIQUE,
   project_id BIGINT,
   cost FLOAT NOT NULL,
   approval_status VARCHAR(50),
   PRIMARY KEY ( purchase_id ),
   FOREIGN KEY (project_id)
        REFERENCES project(project_id)
);

CREATE TABLE IF NOT EXISTS works_on (
      project_id BIGINT NOT NULL ,
    student_id BIGINT NOT NULL,
    FOREIGN KEY (project_id)
        REFERENCES project(project_id),
      FOREIGN KEY (student_id)
        REFERENCES student(asu_id)
);

CREATE TABLE IF NOT EXISTS faculty_makes_purchase(
   faculty_id BIGINT NOT NULL,
   purchase_id BIGINT NOT NULL,
   material_id BIGINT NOT NULL,
   purchase_quantity FLOAT NOT NULL,
   FOREIGN KEY (faculty_id)
        REFERENCES faculty(asu_id),
      FOREIGN KEY (purchase_id)
        REFERENCES purchase(purchase_id),
      FOREIGN KEY (material_id)
        REFERENCES material(material_id)
);

CREATE TABLE IF NOT EXISTS student_makes_purchase(
   student_id BIGINT NOT NULL,
   purchase_id BIGINT NOT NULL,
   material_id BIGINT NOT NULL,
   Purchase_quantity FLOAT NOT NULL,
   FOREIGN KEY (student_id)
        REFERENCES student(asu_id),
      FOREIGN KEY (purchase_id)
        REFERENCES purchase(purchase_id),
      FOREIGN KEY (material_id)
        REFERENCES material(material_id)
);



INSERT INTO material
(material_name,description,sold_per,price,quantity)
 VALUES (
"Vinyl", "Color ($0.55)","linear ft", 0.55, 100
);
INSERT INTO material
(material_name,description,sold_per,price,quantity)
VALUES (
"Vinyl", "Heat transfer ($3.00)","linear ft", 3.00, 100
);
INSERT INTO material
(material_name,description,sold_per,price,quantity)
VALUES (
"Lumber", "2\"x4\" ($0.34)","linear ft", 0.34, 100
);
INSERT INTO material
(material_name,description,sold_per,price,quantity)
 VALUES (
"Lumber", "4\"x4\" ($0.94)","linear ft", 0.94, 100
);
INSERT INTO material
(material_name,description,sold_per,price,quantity)
VALUES (
"Acrylic", "Clear 1/8\" ($3.00)","sqft", 3.00, 100
);
INSERT INTO material
(material_name,description,sold_per,price,quantity)
VALUES (
"Acrylic", "Clear 1/4\" ($5.00)","sqft", 5.00, 100
);
INSERT INTO material
(material_name,description,sold_per,price,quantity)
VALUES (
"Acrylic", "Color 1/8\" ($5.00)","sqft", 5.00, 100
);
```

```
INSERT INTO staff (first_name, last_name, email, staff_password,
role)
 VALUES (
"Renu","Chaw", "rchaw21@asu.edu", "1234", "Admin"
);
INSERT INTO staff (first_name, last_name, email, staff_password,
role)
VALUES (
"Ann","cela", "acela4@asu.edu", "1234", "Accountant"
);


INSERT INTO faculty (first_name, last_name, email,faculty_password,
account_number)
VALUES (
"Usha","Jagannathan", "usha1@asu.edu","1234", 12121212
);
INSERT INTO faculty (first_name, last_name, email,faculty_password,
account_number)
VALUES (
"Damien","Doheny", "damein1@asu.edu","1234", 12121213
);

INSERT INTO student (first_name, last_name, email, student_password,
account_number)
```

```
VALUES (
"Ramya","Chandrashekar", "rchand21@asu.edu","1234", 12121214
);
INSERT INTO student (first_name, last_name, email,student_password,
account_number)
VALUES (
"Mohan","Suraj", "mlakshm5@asu.edu","1234", 12121215
);
INSERT INTO student (first_name, last_name, email,student_password,
account_number)
VALUES (
"Siva","Abhisek", "msiva5@asu.edu","1234", 12121216
);


INSERT INTO project (project_name, description, course_id,
faculty_id, account_number)
VALUES (
"Inventory Management","Innovation hub Inventory Management",
"IFT540", 1, 12121222
);
INSERT INTO project (project_name, description, course_id,
faculty_id, account_number)
VALUES (
"Credit card authentication","card theft tracking", "IFT540", 1,
12121222
);
```

```
INSERT INTO purchase (project_id, cost, approval_status)
VALUES (
1,30,"Pending"
);
INSERT INTO purchase (project_id, cost, approval_status)
VALUES (
2,35,"Pending"
);
INSERT INTO purchase (project_id, cost, approval_status)
VALUES (
1,30,"Pending"
);
INSERT INTO purchase ( cost, approval_status)
VALUES (
30,"Pending"
);
INSERT INTO purchase ( cost, approval_status)
VALUES (
30,"Pending"
);


INSERT INTO works_on (project_id, student_id )
VALUES (
1,1
);
INSERT INTO works_on (project_id, student_id )
VALUES (
1,2
);


INSERT INTO faculty_makes_purchase (faculty_id, purchase_id,
material_id, purchase_quantity )
VALUES (
1,4,1,10
);

INSERT INTO student_makes_purchase (student_id, purchase_id,
material_id, purchase_quantity )
VALUES (
1,5,1,10
);
```