CEN 4010: Principles of Software Engineering

Spring 2018

Milestone 3: Detailed Requirements, Architecture and a Vertical Software Prototype

CEN 4010 Milestone Group 1

FAU Web Store (Perry's Parts Pavillion)

Jasmin Hinton [Scrum Master, Team Leader, jhinton2015@fau.edu]

Richard Altamore [Product Owner]

Jonathan Itty [Development Team]

David Berry [Development Team]

Monday, March 26, 2018

History Table:

1/23 – 1/29: Team Introductions & Role Assignments; Milestone 0

1/29 – 2/19: Picking a Project to do & Researching it; Milestone 1

2/19 - 3/12: Initial GUI Mockup (Framework for Website); Milestone 2

3/12 - 3/26: Implementation of our Search Function; Milestone 3

3/26 - 4/16:

4/16 - 4/30:

Executive Summary

"Perry Parts Pavilion" is a web store we are creating in order to make accessing, retaining and organizing equipment easier for students and staff. Many students with engineering majors will need to take classes that require them to pick up lab kits and other materials from the equipment storage room in the EE building (such as "Logic Design" and "Intro to Microprocessors"). Our Web Store helps make the process of picking up those kits/parts easier for both staff and students. Through our Web Store, customers will be able to order parts/kits (through a simple sign up), rent some of the equipment, request permission to use some of the more advanced equipment if necessary (such as the laser cutter or 3D printer), and a number of other tasks will be possible. Even with all the capability of this store one of the overall goals for it is to make it as user friendly as possible, while still allowing for optimum efficiency. One of the main benefits of this Web Store is its simple accessibility.

A Web Store available for the students to use at their own discretion allows the department to accurately keep track of equipment and parts. This Web Store will be able to monitor everything that happens in the equipment storage room. The students merely have to sign up, place an order, and pick it up (for the ordering a part/kit feature). Thus, allowing the department and suppliers to kept stock of what is in stock and what may need replenishing. Along with this is gives students the ability to view all parts they may possibly need for a given project. Rather than combing through a room full of parts all parts necessary can be seen on one screen. The store also allow for accountability and guarantee of product once it has been requested. The departments ability to keep track of how many parts are taken by which students will allow for an understanding of if individuals are abusing the system. Also, with the ability to

track inventory the sight can inform a student when a part is out of stock, rather than having the student search for the nonexistent part.

Having a Web Store implemented helps not only the department but also the student and any suppliers that may be given access to the system. The difficulties of getting parts would essentially be removed completely from the students lives and the worrying about managing all that equipment in the supply room would all be taken care of by our Web Store, as it ensures that it is aware of everything that goes on in that room. Florida Atlantic University's CEECS department would benefit greatly from the use of our Web Store and we, as students, would also be able to benefit from it ourselves. The FAU Web Store is the way of the future for that equipment supply room in the Engineering East building.

Competitive Analysis

The two main forms of accessing that equipment supply room, and in turn, the equipment inside it, are presently:

1. Going to the supply room, looking for the parts/kit that you need for your class, and then answering questions regarding who you are and what you need it the requested parts for. This method is currently being implemented.
2. Using our web store to easily place any order, rent any equipment, request any machinery use, etc. This is the method that we are proposing and the method we want to be implemented into the system.

Using the Old Method to get your Kits/Parts	You must go to the supply room.	It takes a while to look for the parts you need.	You get your parts and need to sign for them.	Keeping track of all the parts that get rented out is difficult.	Human error could lead to losing parts.	Time slots for using advanced machinery is difficult to track.
Using the New FAU Web Store to get your Kits/Parts	Easy to order when your not at FAU.	You are given an exact location for the parts.	You get your parts and go without delay.	Keeps track of people who rent parts and sets a time limit.	The Web Store keeps track of all parts.	Appling to use the advanced machinery is simple and easy.

Looking at what the system is using right now, we can see there are certain aspects of it that can be improved drastically. The major examples are listed above. The most important would have to be the first, going to the supply room, this being because it does not accommodate

for students who commute to FAU. These students have difficulty finding time in their schedules to make it in person to the supply room. The option of a web store allows for these students to have guaranteed parts before making the trip to school . With the Web Store in place, they could easily order their parts and quickly stop by FAU, and pick them up. With the Web Store, students would also have a much easier time finding the parts they're looking for, as well as spend less time in the supply room in general. The student would place the order on the store, get the location, and from there it's a simple get in and get out. Renting parts and keeping track of their whereabouts is also made a lot easier due to the fact that the Web Store takes care of it all. Along with this, the time slots for using the advanced machinery are unpredictable, but with the Web Store it's all made clear. These are just some of the many benefits of using our Web Store.

Data Definition

STUDENT: Holds accounts through website. Is able to rent PRODUCTS. Has a COURSE LIST.

STAFF: They are a step below ADMIN, and their job is to manage the STUDENTS and keep track of the PRODUCTS.

ADMIN: They are a step below SUPER ADMIN, and they are able to view the PRODUCT, view COURSE LIST, etc.

SUPER ADMIN: Perry, who has total access to everything on the website. Is able to view STUDENTS, view COURSE LIST, change PRODUCTS, etc.

LOGIN PAGE (page 1 on Website): the page that allows STUDENTS, STAFF, ADMINS, and SUPER ADMINS to login.

CREATE AN ACCOUNT (page 2 on Website): This is where, if you don't have an account you would create one so that you can use our WebStore. You would fill in your Name(First, Last), Z Number, Email, Phone Number, Password, and Re-enter Password.

CUSTOMER PAGE (page 3 on Website): The main page of our Website where the user is able to use SEARCH BOX for items, ORDER PARTS, REQUEST ITEMS, REQUEST JOBS, or RENT ITEMS.

SEARCH BOX: allows the user to specify what exactly they are looking for on our Webstore.

ORDER PARTS: allows the user to order specific parts that they might need in their classes. [Falls under the PURCHASE REQUEST category]

REQUEST ITEMS: allows the user to ask for permission to get an item. [Falls under the PURCHASE REQUEST category]

REQUEST JOBS: allows user to ask for permission to do a JOB. [Falls under the JOB category]

RENT ITEMS: allows the user to rent out a part temporarily, should they need that specific part for an assignment in their class.

INVENTORY PAGE (page 4 on Website): hows the list of everything that our Webstore has to offer, in terms of our parts and EQUIPMENT. Is has the Name, Part Number, Description, Quantity, and Price.

COURSE LIST: A descriptor of STUDENT, telling what classes the student is in for COURSE REQUIREMENTS on EQUIPMENT and TIMESLOTS

PRODUCT: Can either be EQUIPMENT, a JOB, a PURCHASE REQUEST, or a TIMESLOT. STUDENTS will request access to PRODUCTS through the web page.

EQUIPMENT: A part that can be rented from Perry's lab by a STUDENT and taken home to be used for classes. Has a LOCATION, COURSE REQUIREMENT, and a STATIC RATING. Will

be shown on its own page with an image and description of the item. Equipment can be packaged into KITS.

JOB: A job to be done such as 3d Printing something. Must give a reason for needed job to be done.

PURCHASE REQUEST: A request for the lab to purchase a certain part for STUDENT use. Has a PRICE and AVAILABILITY.

TIMESLOT: A time for a STUDENT to use a piece of advanced machinery available. Has a time and COURSE REQUIREMENT.

KIT: STUDENTS can create a KIT from parts in the lab. Is a collection of EQUIPMENT and PURCHASE REQUESTS. STUDENT must meet all requirements of each piece of EQUIPMENT for renting, or must make a special request.

LOCATION: A descriptor of EQUIPMENT, telling where on Perry's shelf the piece is. If the part is being rented, it will be listed here and will be unrentable by other students until returned.

COURSE REQUIREMENT: A descriptor of EQUIPMENT and TIMESLOT, telling what class the STUDENT must have in their COURSE LIST in order to rent this part or use this machinery. Special request can be made if the course is not being taken by the STUDENT.

STATIC RATING: A boolean descriptor of EQUIPMENT, telling if it is static sensitive or not. Will need to be covered in pink foam if true.

PRICE: A descriptor of PURCHASE REQUESTS, telling the cost of the item that is being requested

AVAILABILITY: A boolean descriptor of PURCHASE REQUESTS, telling if the item is available for purchase from a vendor.

Overview, Scenarios, and Use Cases

The FAU Web Store is being created for the sole purpose of making the process of getting equipment from the EE equipment supply room easier for both students and staff, alike.

The project we are creating is a web store, which will allow users to create an account, order parts, request jobs (such as 3D printing, printed circuit boards, and laser cutting), and rent items. This web store is a model of some similar sites like Amazon, Walmart, Home Depot, etc. The user will be able to simply maneuver the site, by first creating an account with his/her Z number, email, phone number, and other pertinent information, to start their request. If a user decides to order parts then they will be able to select the individual parts, or kits if desired. Initially each part will display a description, image, part number and price. When the parts are low or not in stock it will be displayed for the user, and the vendor will be notified to restock the item. In addition, the admin users are key in updating the site with new inventory, new products, create kits, and able to have full access to all user accounts. The store is intended to be simplistic, and easily accessible for all user levels. This application is supposed to become a way to keep account of inventory, rentals, and kits for classes such as Logic Design and Microprocessor.

High-Level Functional Requirements

Priority Levels:

1- Must Have

2- Desired

3- Opportunistic

1. (Priority 1) User input of information: User will submit form with phone number, email and Z number to create an account. This information will be stored on the SQL database and be used for future logins.
2. (Priority 1) User Search: User is able to search for the desired product or rental availability. User can search by Part Number, SKU or Name. All of the data is stored in the SQL server and is accessed with PHP
3. (Priority 2) User select/request of product: User is able to select parts on the display screen to learn more about them. When user clicks on an item, they are linked to a page with information on the respective item and there will be a button to add this item to the cart if desired.
4. (Priority 2) User request for machinery usage: User can fill out form to reserve machine time. Every time the user completes the form, it uploads the information to a spreadsheet that holds all the information. This portion will be updated on the mockup as it was not in Milestone 2.
5. (Priority 1) Add/Delete Cart Item: This function gives the user the ability to add and delete parts off of their list. The user can also manipulate the quantity of each item they need.

6. (Priority 3) Inventory Tracker: When user completes process of reserving said item the number of items is deducted from inventory. If product in cart becomes unavailable during processing time error message will show indicating the product is no longer in stock.

7. (Priority 3) Rental Date: Rental date is shown and return date is displayed along with a notification of a the number of days (3 or less) in for rental. These products are also deducted from the inventory count to indicated their availability.

List of Non-Functional Requirements

1. Security: Users will login using there FAU username and FAU password. The user's credentials must be current. If the user changes their username/password, the web store must be able to allow the user to login using their new credentials, while also preventing them from logging in using their old credentials. Ideally, this behaviour should be near instantaneous. Also, only those with valid FAU accounts should be given even basic access to the web store.
2. Expected Load: Given that the only users of this system will be Engineering/Computer Science students, plus faculty, the system is not expected to need the ability to maintain and large number of concurrent users. Thus, the expected number of concurrent users is the size of a large classroom: one hundred and twenty users.
3. Accessibility: The interface must be user friendly and relatively simple for customers. Usage of the system should require only a basic understanding of websites and computers. Directions must be clear and concise, without using any engineering lingo. Inclusion of small tutorial windows explaining how to operate certain parts of the system (where to find one's FAU username, for example) are a possibility.
4. Availability: The application must be available to customer level users any time there are classes taking place. For administrator level users, the system should be available any time so long as the server is running. Administrators will be able to open and close the web store.

5. Response Time: Ideally, response time from the system should be under ten seconds. At most, the system can take a minute to respond, assuming there is a stable connection between the user's terminal and the server.
6. Reusability: Given that the system will allow administrator level users the ability to add and remove items from the storefront, the system should be able to remain in use for as long as the server's hardware does not change.

High-Level System Architecture

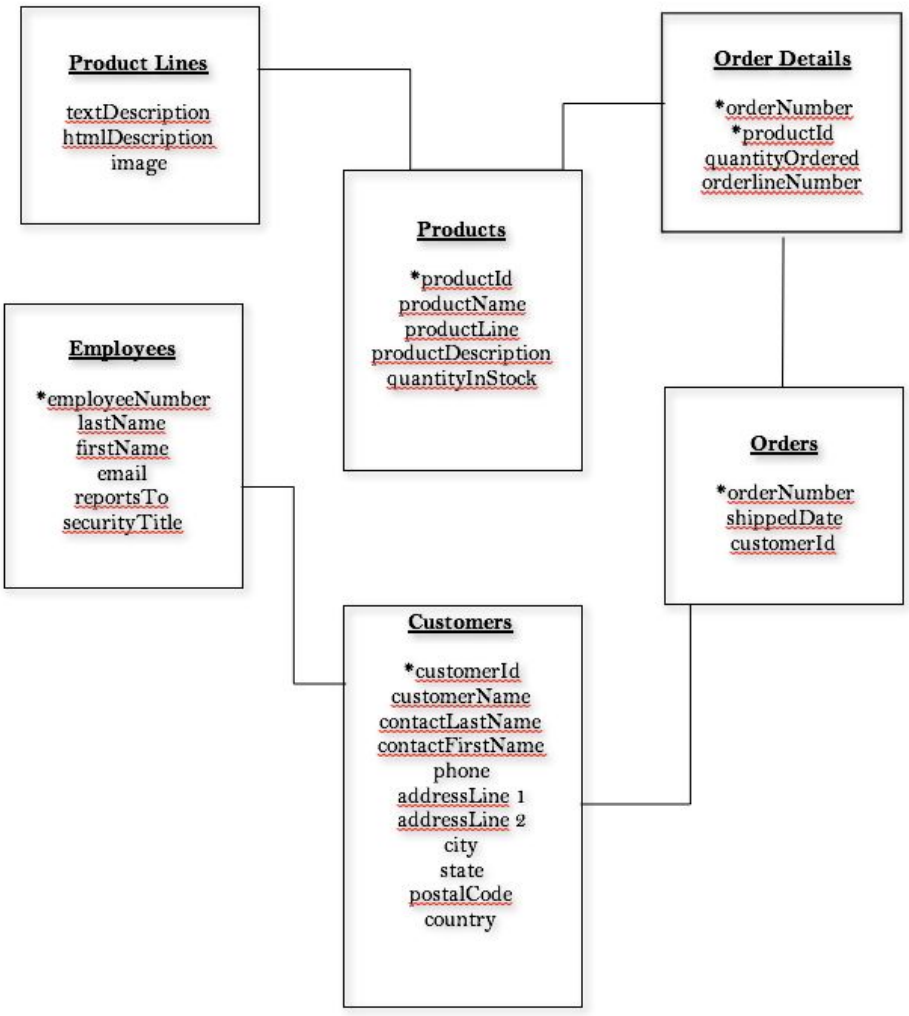
1. High level Architecture of the code
    a. **Web User:** This user will be prompted with a login screen similarly to the outline of our Milestone 2 mock-up. The user will have to enter their login-id which is their Z-number which will be implemented through the use of a string. In addition the password, will be implemented through a string of no specific max characters. Each web user has different states of access which is outlined in the UML diagram below.
    b. **Customer:** If the user does not have a login created then they will have an option displayed on their screen to do so. No purchase requests are able to be completed without having created a login account. The user/customer will be asked for their id (Z-number), address (implemented by string, phone number (implemented by string) and email address (implemented by string).
    c. **Account:** The user has an option to close their account if need be, maybe if they have completed the course and no longer need to go to the Parts Lab for materials. This account process is on a case-by-case basis.
    d. **Line Item:** This section will prompt and display as a pop-up of the quantity of the item the user has selected. Later on, we may change this feature to being a second screen instead of a pop-up.
    e. **Product:** This section will display the name, part number (id) of the product in the form of a string. The quantity will be an integer and the description will display all the pertinent information regarding the product itself. The description

will be its on function, being that we will have multiple products, some of which will have similar descriptions.

f. **Order:** The order section will display the customer-id in the form of a string (Z-number). Also display the date the product was ordered (Date function); this will be a function because it will be used multiple times throughout the program. This function will keep track of when the ordered was made and be able to present an estimated time the order is ready for pickup. Once this estimated time is the delivered the OrderStatus function will be called and an email will be sent out to the user letting them know their order is ready for pickup at the EE lab. The final integer value will be total which will display the total number of items the user has available.

g. **Shopping Cart:** The shopping cart will display when the cart is created with the Date function, it will also display the total number of items in the cart as an integer value. In addition it will display when the user selects order placed, a pop-up window will appear saying “Thank You for placing your order: confirmation email has been sent to you”.

2. **Database Organization**



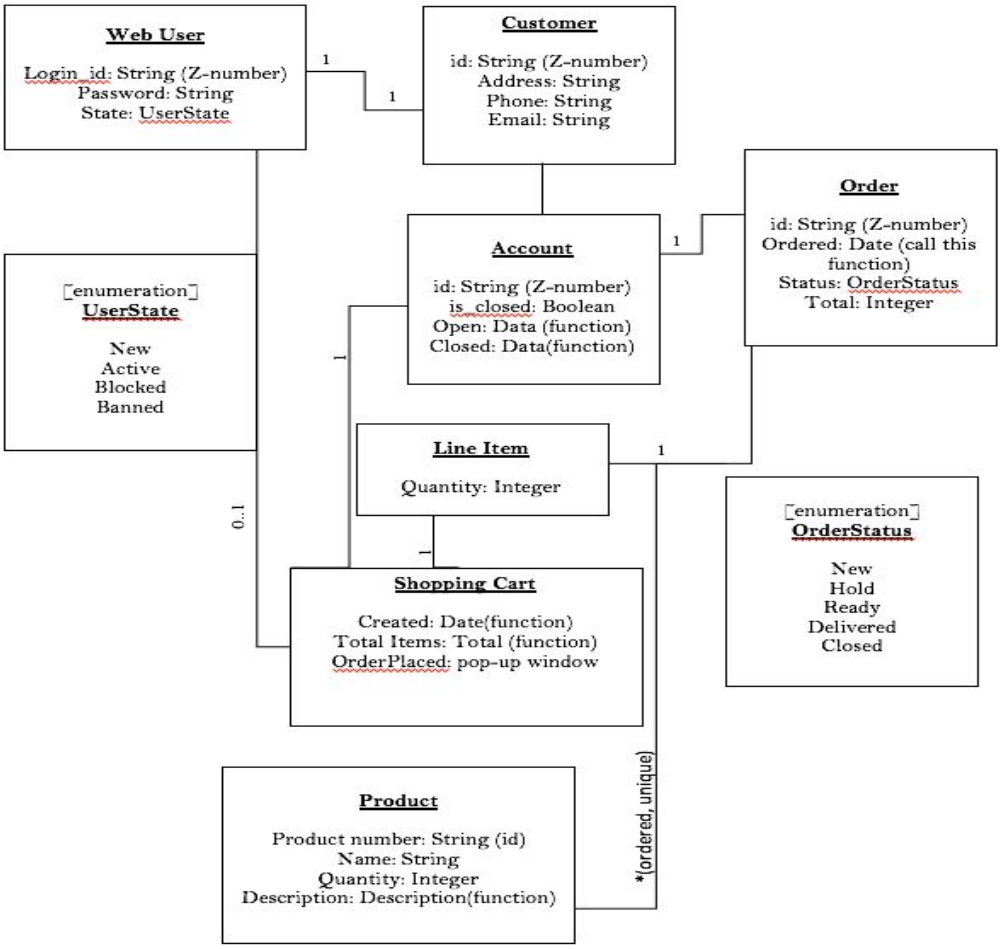
3. Media Storage such as images will be kept in database storage at the moment, if we decide to change this we will update in the final project.
4. At the search entry screen, the user will be presented with a text box and a drop-down menu containing most of the columns in the ‘inventory’ database; thus, the user will be able to search any of these columns for an entry that contains the string that the user types into the text box. As to the search algorithm itself, there are two main components: the search string parsing and the search column name parsing. First, the column parsing:

given that the column names in the database may be different and/or be changed at some point in the future (and to help prevent errors resulting from typing the column names incorrectly), the search algorithm first creates an SQL statement that searches for a column name that contains the keyword associated with its selection in the search drop-down menu. For example, there are two categories in the sample database that contains the phrase “p / n”: “P /N Name and Description” and “Newark P / N”; therefore, the keyword for the first column containing said phrase would be ‘name’, while the keyword for the second column would be ‘newark’. Once the SQL server returns a table of the names of the columns that contain the keyword, the algorithm selects the first entry in said table and assigns it to a variable to be used in the second part of the algorithm: the search text parsing. The search text parsing is simple: it creates an SQL statement that will order the server to return all entries in the database that contain the relevant string in the column that the user indicated. However, for a request to search for “keywords”, the algorithm as a special SQL statement to search both the “Keyword1” and “Keyword2” columns. Regardless, the algorithm then prints out the entries that match the search conditions to a webpage for the user’s perusal. In order to get the correct column names for display, the algorithm uses functions that are similar to the first part of the search algorithm.

5. The primary APIs that will be created for the webstore will be for communication with the mySQL server. Also, if this project to meant to be fully functional, APIs governing connections to various bank servers would also be needed.

6. We have no significant or non-trivial algorithm or process to add at the moment, if need be will adjust in the final project display.

High-Level UML diagrams



UML Component and Deployment Diagram

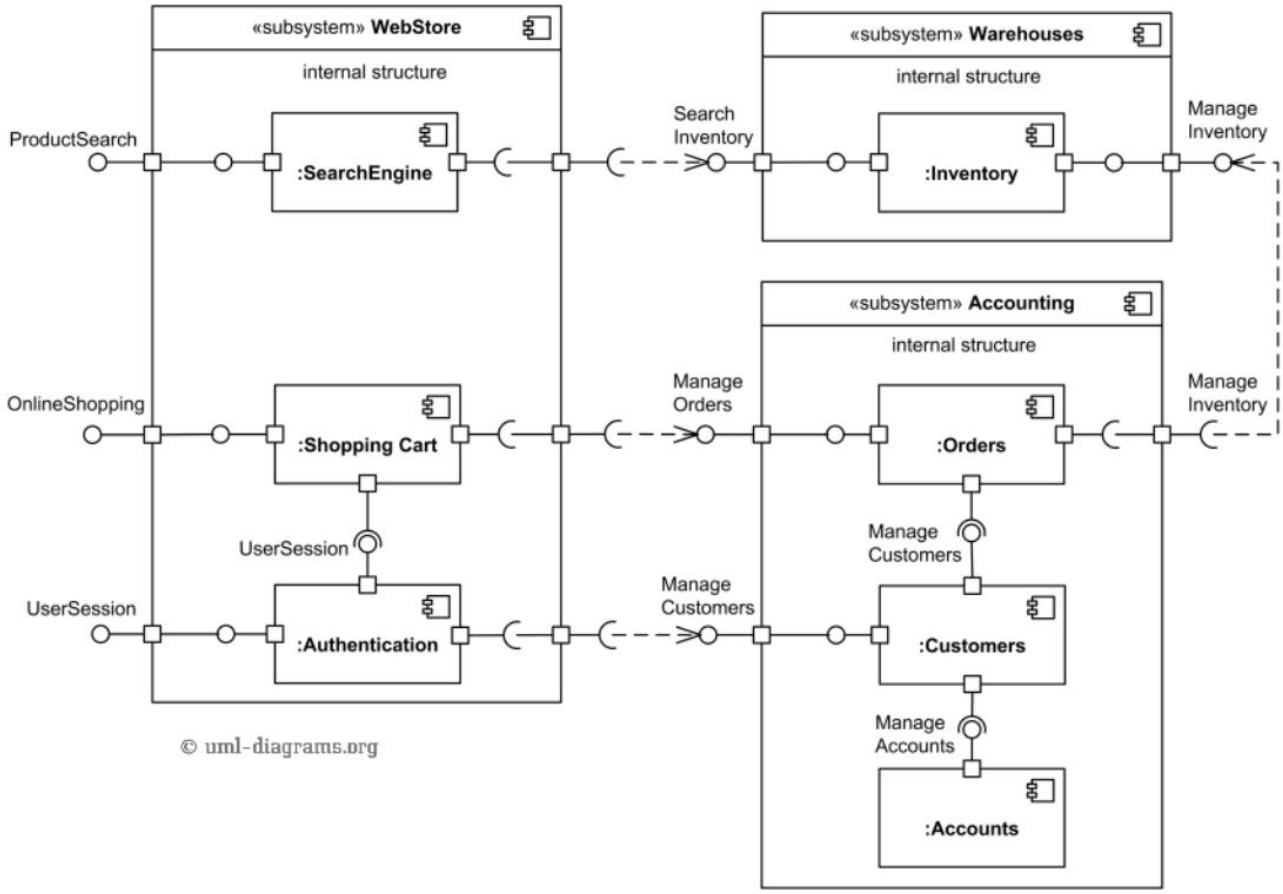


Fig. 1: This is a sample UML deployment diagram we will be using.

Identify Actual Key Risks For Your Project At This Time

Skill Risks:

A majority of this project mainly focuses on us dealing with programs/topics that we are unfamiliar with. As such, it is clear as day that most of us do not have enough skill (in whatever section we are working on at the time) to be able to work in an efficient and productive manner. To compensate for this major problem, much of our time is spent researching how to use a program correctly so that we may be able to utilize it and discuss it properly when it comes to making our project.

Schedule Risk:

In terms of schedule risks, we are not experiencing issues. On top of that the lack of materials or tools to complete this project is the main difficulty. We might run into several problems later on depending upon how well we handle the rest of the implementation. Schedule risks are closely related to technical risks because of the fact that any unforeseen problems that we run into (categorized as a technical risk) may impact our schedule based on how fast we can get them resolved.

Technical Risks:

In terms of technical risks, we are not experiencing any at the time. Our development team knows what is going on in our project and they are fully capable of dealing with whatever comes along. However, we have only scratched the surface of our implementation plan. We may run into several problems when we begin to implement some of the more difficult functions.

Teamwork Risks:

In the beginning, we seemed to lack the proper teamwork in order to get this project done, but by the time Milestone 1 came, we had essentially solved this problem. We all were unsure of what was going to happen in the future, since we had lost one of our members, Ryan. At the beginning, Ryan was the initial leader, so when we lost him we were kind of in trouble. But our Scrum Master / Team Leader, Jasmin, was able to fix our poor teamwork and lead us properly. She picked up the role of team leader as well as Scrum Master so that we could be successful in this project. Everything is okay now and it actually works out really well too, since the Scrum Master's role was to ensure that we are all on top of our specific tasks.

Legal / Content Risks:

In terms of legal / content risks, I do not believe we are at risk at all. We have been extremely careful not to copy any outside sources, so in terms of content, it is purely our own idea. Legally, I don't believe that we have violated any laws when it comes down to creating out website project so we should not have to worry about this risk.

Team

Scrum Master --- **Jasmin Hinton**

Handles time management and oversees the progress of the development team so that they do not deviate from the Scrum method and its values.

Product Owner --- **Richard Altamore**

Handles portions of the paperwork and any and all new features that might be added to the Web Store.

Development Team --- **Jonathan Itty**

Assistant to the lead programmer of the team, David. He will be tasked with helping out on the coding aspect of the Webstore.

Development Team --- **David Berry**

The lead programer of the team, has the most advanced knowledge of how to setup the basis for the Web Store we are creating. He is charged with the handling of the major portions of the coding.

Checklist

a) Team decided on basic means of communication: **DONE**

b) Team found a time slot to meet outside of class: **ON TRACK**

c) Front and back end team leads chosen: **DONE**

d) GitHub master chosen: **DONE**

e) Team ready and able to use the chosen back and front-end frameworks: **ON TRACK**

f) Skills of each team member defined and known to all: **DONE**

g) Team lead ensured that all team members read the final M1 and agree/understand if before submission: **ON TRACK**