CEN 4010 Principles of Software Engineering, Spring 2018.

Milestone 3 Project Proposal and High-level description

Date: FEB 19th, 2018

Team Name: ELEVEN, PROJECT ELEVEN.
Team Number: 11

Roster:

- Thiago De Mendonca,
  - contact: tdemendonca2017@fau.edu

- William Tring
- Roberto Beltran
- Piotr Skipor

2. Executive Summary

The following is a proposal for a Principles of Software Engineering Project: ELEVEN.

A web store solution for a system environment consisting primarily of distribution and inventorying of hardware parts. Our goal is to offer a robust and user friendly solution for small businesses and corporations in the process of migrating to a digital system.

Although we have experienced incredible growth in the digital sector over the years, many small businesses and operations still are heavily invested in physical data collection. ELEVEN is a solution that aims to eliminate issues such as user errors going unchecked, inventory being a major undertake, increased costs associated with physical records and data collection and data retrieval being inefficient.

The ELEVEN project will be web based, therefore eliminating the need of a specific local program or hardware for both user interface and administrators alike. The system shall have a dedicated digital database with expanding abilities for inventory and users, a user friendly interface, a list of offered services such as equipment rental and task oriented requests, use of API for inventory tracking, security improvement and data safety.

Implementation will be done continuously and in stages. We will first develop and test the basic components. The following stages will be to

integrate the databases, API's and additional services. The final stage will be to implement and validate the software.

This system could lead to future projects involving more extensive databases, as well as increased performance and optimization. Therefore it will not only provide and excellent solution initially but will carry the growth potential to accompany the environment it is implemented with.

3. Competitive analysis

Analyzing competitive products available today. Present competitors' features vs. your planned ones. First, create a table with key features of competitors vs. yours. Only at very high level, 5-6 entries max. After the table, you must summarize what are the planned advantages or competitive relationship to what is already available.

Our Store

Other Online stores

Already assembled set ups of bread Boards	Pictures of ready Products
Search for any part in the store	User reviews of products
Location of each product in the store	User friendly website
Pictures of how to assemble bread	Easy shipping

boards	
Large Data base	More verity of finished products

There are many online stores where we can buy electronics parts however those website do not have kits that are already assembled for each classes (Logic design, Microprocessors ,Engineering Design)
Our Online store will provide hundreds of electronic parts and packages for each computer engineering hardware classes.

4. Data definition

This section serves as the “dictionary” of your document. It defines main terms, data structures and “items” or “*entities*” *at high or logical (not implementation) level* (e.g. name, meaning, usage, and NOT how the data is stored in memory) so it is easier to refer to them in the document. Focus on key terms (main data elements, actors, types of users etc.) specific for your application and not on general well know terms. These terms and their names *must be used consistently* from then on in all documents, user interface, in naming software components and database elements etc. In

later milestones, you will add more implementation details for each item. You will later expand this section with more details.

Data Entry

: Inventory –

Logic Boards packages: Logic Design Package, Micro processors Package, Engineering design Package.

Parts (store inventory) :

9V Battery Connectors

Battery 9v

LED, Red, 5mm T-13/4

Servo sg90

DC Motor

Ping Sensor

Temperature Sensor

Resistor

PIR sensor

GL5516 CdS Photo Cell

Microphone

Soldier Proto Board 5cm*7cm

Bread Board

Ardinuo Nano USb

PCB Milling (sq in)

Jumper wires M-F 20psc

Project Container

5. Overview, scenarios and use cases

This section describes the project overview (in much more details) and likelihood usage scenarios of your product from end users' perspectives. Focus only on main use cases. Simple text format is OK and preferable – tell us a story about who and how is the application used. Focus on WHAT users do, their skill level, not on HOW the system is implemented. You can expand use cases provided in high level document in future milestones.

Customers may only browse the store. If they wish to access other user features, such as purchasing, they could create an account by signing up with their name, z-number, email, college, department, graduation date, classes taken (3 letters, 4 numbers), and password. Afterwards they may login to gain access other user features.

Customers would be able to access the website and look for products or kits which they wish to purchase. They could either browse through the website by navigation or search directly for what they want. After they find the item, they could look at the product description and choose to add the item to the cart or download related datasheets and documents for the product. The product would have a different price for single purchase or bulk purchase. Once the products are in the cart, the user can choose to either rent the product, if applicable, or purchase it in which case an amount is credited from the user's account. The website also asks users what the purpose of the purchase is, whether it's required for a course or not.

If a customer cannot find an item, they could submit a request for the store to add the product to the inventory list.
Customers can also request a job from the store to use the 3D Printer, Printed Circuit Board, or Laser Cutter. They must upload the correct files for each job and have an option to leave a comment as well. The job would be queued up for a employee to complete. The user is given a order ID to track their job request. An alert would be sent to the store for an employee to complete the job order.
Employees create their account by signing up the same way but entering a staff key ID which grants them permission to access employee features after logging in.
Employees could access the store the same way a customer could. They may also edit products such as their descriptions, prices, or availability. Employees could add or remove products to the website. They could form new kits/bundles with existing products to create a new product.
Employees could reorder more inventory through available vendors.
Employees could view customer information and details about their purchases. Purchase orders would be displayed organized by where the inventory is located to make it convenient for employees to gather and pack the products.
Admin accounts has access to everything an employee does as well as being able to approve new item requests.
Admin may edit customer and employee accounts. Admin could grant/reject permission for employees to access certain functions such as being able to order inventory. Admin may add or remove vendors.

6. High-level functional requirements

This refers to the high-level functionality that you plan to develop to the best of your knowledge at this point. Focus on WHAT and not HOW. Keep the users in mind. Develop these functions to be consistent with use cases and requirements above. Number each requirement and use these numbers consistently from now on. For each functionality use 1-5 line description.

Functional requirements listed by user cases:

Customer

Priority 1:

Sign-up (Customer/Employee) - create a user profile with login details, user information, what kind of customer (teacher/student) or employee.
Login (All Users) - create session ID and access to more features depending on type of user account.
Search - search for products or kits by using keywords, if not found then offers to request the item.
Navbar - for website browsing.

Priority 2:

Product Details/Docs - product description box and related downloadable documents such as datasheets.

Cart/Checkout - add to cart option and purchase. Credits the user account, asks for reason of purchase, provides order ID.
Item Request Form - provides user with a form to submit to admin.
Job Request - different job request page for each type of job. Allows user to upload documents and images as well as leaving comments for the job.
View Order Transaction - shows order details and tracking ID.

Priority 3:

View Profile (All Users) - shows account information and order history. Allows user to change their own profile. Employees can see customers' login information also. Admins can see employee information and edit all user information.
Product rating system - users can rate products and add personal insight to items with a description of their experience.

Employee

Priority 1:

Order Inventory - shows inventory and allows employees to order more inventory from vendors. Displays different prices from different vendors.

Priority 2:

Edit Products - can add or remove products and edit product details.
Data sorting and inventory sorting options - sort by: date added, price, alphabetical order, model, SKU number, availability.
View Job Requests - employees can view job requests and everything attaining to the request.

Priority 3:

Organize Purchase Order - purchase orders are displayed organized to employees for convenience.

Admin

Priority 1:

View Item Requests - displays item request messages for the admin.

Priority 2:

Add/Remove Vendors - displays list of vendors and allows admin to make changes based on vendor availability of items and economical value offered.

Vendor accessibility for inventory management and price comparison.

Priority 3:

Edit/View users data log and request logs.

7. List of non-functional requirements

For example, performance, usability, accessibility, expected load, security requirements, storage, availability, fault tolerance etc. Number each. When possible, try to quantify these quality attributes.

1. Usability
2. Availability

3. Accessibility
4. Performance: response time, throughput, utilization.
5. Dependability
6. Security requirements
7. Expected Load
8. Storage
9. Fault Tolerance
10. Regulatory requirements
11. Ethical requirements

8. High-level system architecture and database organization

Lists of main software products, tools, languages and systems to be used, list of core APIs available at this point, supported browsers etc.
You also have to decide on which frameworks you will use if any. These provide both user interface, as well as cross-platform and cross browser layout/css. All external code you plan to use must be listed along with their license.
(SQL, HTML, PHP, Java… Chrome, IE, Bootstrap)

Back end: MySQL and Java
Front end: Brackets, HTML, CSS, Python
Other: Internet explorer, Chrome, Mozilla, Microsoft edge, GitHub, Canvas, MEETS, WhatsApp

1) High level Architecture of the code must be consistent with UML class diagram (see below).

user case: user will be prompted to create an account or sign in as pointed in milestone 2. User will enter their login id and password, depending on the login characteristics user will have different levels of data access.
customer: In order to complete a purchase or an job request the user must be logged in with proper credentials using their znumber as main form of identification.
account: User will have an account log and will have the option of closing existing account and check account status for unreturned items or pending fees.
Inventory: This query based system will keep track of all items using parts numbers to identify availability and provide product description.

2) DB organization: Describe the main database schema/organization (high level), e.g. list main DB tables and items in each DB table

User info (table): Name, znumber, class-crn, college, department, class, classnumber,
classname, email
Inventory (table): SKU, P/N Name/Description, keyword1, keyword2, Newark p/n,
quantity, price each

3) Media storage: Decide if images and video/audio will be kept in file systems or in DB. Describe any other special data format requirements like for video/audio/GPS etc.

All media files will be stored in the database on the database server. We can use default database to
store the media or we can create a new one. In order to use new database as our media storage we
have to add information about it and its access credentials to the local.xml file.
How does it work ?
In order to get files from the database to the file system on the web PHP script is used get.php
Web server rewrites are disabled in the Magento system and the PHP script is running and then required
media that exists in the database is pulled out.

4) Search/filter architecture and implementation: what will be the algorithm for search; what DB terms will be searched, how it will be coded and organized in the DB. Similarly, say what DB items will be filtered/sorted

Sorting the database
Search bar will lead to a php page to show results. Search will search products by using
their keywords as tags. Results will be shown based on most similar first. We can then
sort by price or quantity.

```
switch (sortOrder)
  {
     case "Part number":
        parts= parts.OrderByDescending(s => s. number);
        break;
     case "Date":
```

```
        parts = parts.OrderBy(s => s.dateOfPartAdded);
        break;
      case "Part_size":
        parts = parts.OrderByDescending(s => s.partSize);
        break;
      default:
        parts = parts.OrderBy(s => alphabeticalOrder );
        break;
    }
    return View(parts.ToList());
}
```

5) Your own APIs: Describe and define at high level any major APIs that you will create

Create will create a new customer
Info will show us data on specific customer
Update will update data for specific customer
Remove will delete all data for specific customer.

```
<config>
    <api>
        <resources>
            <customer translate="title" module="customer">
                <title>Customer Resource</title>
                <methods>
                    <list translate="title" module="customer">
                        <title>Retrive customers</title>
                    </list>
                    <create translate="title" module="customer">
                        <title>Create customer</title>
                    </create>
                    <info translate="title" module="customer">
                        <title>Retrieve customer data</title>
                    </info>
                    <update translate="title" module="customer">
                        <title>Update customer data</title>
                    </update>
                    <delete>
```

```
            <title>Delete customer</title>
          </delete>
        </methods>
        <faults module="customer">
        </faults>
      </customer>
    </resources>
  </api>
```

6) Describe any significant non-trivial algorithm or process (like rating, ranking, automatic prioritizing of items etc.)

We will use A ranging algorithm in order to sort more recent parts that are entered to the data base.

I will use Reddit ranking algorithm.

Reddit algorithm is implemented in Python. The sorting algorithm uses Pyrex in order to write Python C

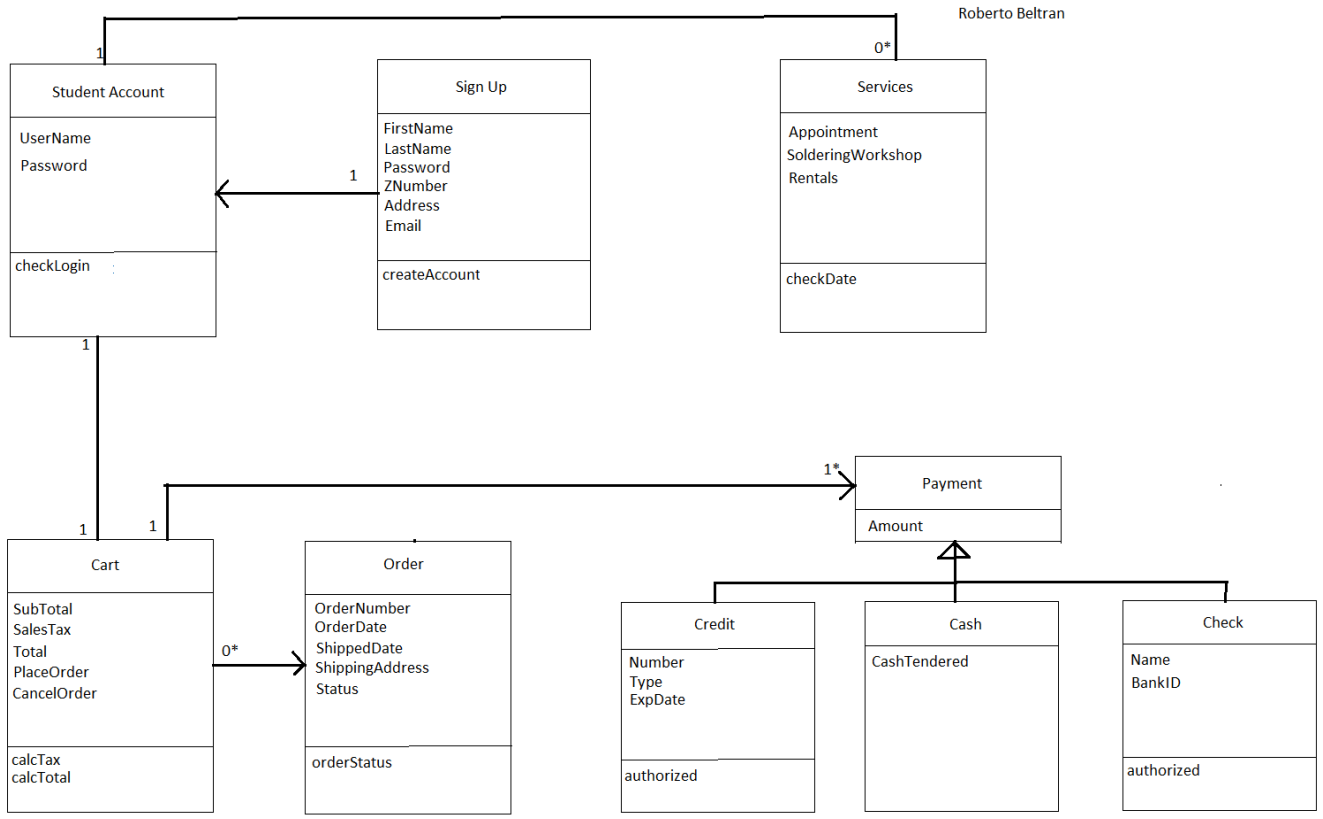
extensions.

Submission time is the key to ranking and the algorithm will rank newer parts higher than older.

The score will not decrease as the time goes by however, newer parts will get higher score than older

and they will be automatically shifted to the top.

9. High-Level UML diagrams



10. Identify actual key risks for your project at this time

Identify only actual and specific risks in your current work such as (list those that apply:

1) Skill risks is our main issue, code implementation and workload completeness are an issue at the moment despite all members working to the best of their abilities.

2) We were having some schedule risks. Due to most of us having multiple classes and work, we had trouble finding a time to meet up that works for all of us. We have come to an agreement to meet every other Thursdays at 10:30 am.

3) We do not have any Technical risks.

4) We have a Teamwork risk. Some of our team members are not communicating as much as we would like. Tasks aren't being tackled until the last minute creating unnecessary stress and risks.

5) We do not have any Legal/content risks

HISTORY TABLE

contents will go below this point.