**CEN 4010 Principles of Software Engineering, Spring 2018.**

**Milestone 1 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

# 

# **1.** **Executive Summary \*\*by Thiago De Mendonca**

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.

# **2** **Competitive analysis \*by member Piotr Skipor**

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 boards | Easy shipping |
| 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.

# 

# 

# **3** **Data definition \*by member Piotr Skipor**

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

# 

# **4** **Overview, scenarios and use cases \*by member William Tring**

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.

# **5** **Initial list of high-level functional requirements \*by member William Tring**

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.

Customer  
Search - search for products or kits by using keywords, if not found then offers to request the item  
Navbar - for website browsing  
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.  
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 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  
View Order Transaction - shows order details and tracking ID  
  
Employee  
Edit Products - can add or remove products and edit product details  
Order Inventory - shows inventory and allows employees to order more inventory from vendors. Displays different prices from different vendors.  
Organize Purchase Order - purchase orders are displayed organized to employees for convenience  
View Job Requests - employees can view job requests and everything attaining to the request  
  
  
Admin  
Edit all users  
Add/Remove Vendors - displays list of vendors and allows admin to make changes  
View Item Requests - displays item request messages for the admin

# **6** **List of non-functional requirements \*\*by member Robert Beltran**

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  
5. Dependability   
6. Security requirements  
7. Expected Load  
8. Storage  
9. Fault Tolerance  
10. Regulatory requirements  
11. Ethical requirements

# **7** **High-level system architecture \*\*by member Robert Beltran**

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

# **8** **Team by Thiago De Mendonca**

List student group names, name of Scrum master, product owner and initial roles for each member

Product owner: Perry Weinchall

Initial Team Roles:

Team Lead: Thiago De Mendonca

Scrum Master: Piotr Skipor

Team Meeting Schedule: Thursday's (9am-10am)

Front & Back End Leads: Thiago De Mendonca and William Tring

GitHub Master: Thiago De Mendonca

# **9** **Checklist by Thiago De Mendonca**

For each item below you must answer with only one of the following: DONE, ON TRACK (meaning it will be done on time, and no issues perceived) or ISSUE (you have some problems, and then define what is the problem with 1-3 lines)

a) Team decided on basic means of communications: DONE

b) Team found a time slot to meet outside of the class: DONE

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: ISSUE

-Pending on additional info from Perry eg. API.

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

-Have yet to establish each one's abilities to properly commit on sprints.

g) Team lead ensured that all team members read the final M1 and agree/understand it before submission: DONE

# **10** **Tasks before submission \*\*by Thiago De Mendonca**

Teams must collaborate in creating M1 document by having working M1 document on their team GitHub private repository (similar to managing code) so all team members can access it. Added advantage of doing it this way is that it builds teamwork and communication. We recommend having a folder for project documentation on team’s GitHub where milestones and other similar files can be kept.

-TASK IS DONE

# **11** **Submission \*\*by Thiago De Mendonca**

Each team submits one single word document with all the above required sections to Canvas by the due date. Must have a title page to your document, including:

a) Course Title and term: CEN 4010 Principles of Software Engineering, Spring 2018,

b) Document name: Milestone 1 Project Proposal and High-level description

c) Your team name, and project name (you can use the name you chose for your team)

d) Team number (I will assign you one)

e) Names of students (team lead first) with e-mail of team lead

f) Date

g) History table (revisions) (Note: you will update this document based on instructors’ feedback so this is important)

**HISTORY TABLE**

contents will go below this point.