**Door Production Manager**

**Table of Contents**

**i. Change Log……………………………………………………………………………….………...2**

**Ii. Project Proposal………………………………………………………………………….………..2**

**iii. Glossary…………………………………………………………………………………………….3**

**iv. Functional Requirements…………………………………………………………………….….4**

**v. User Types………………………………………………………………………………………......5**

**vi. Data Objects………………………………………………………………………………………..6**

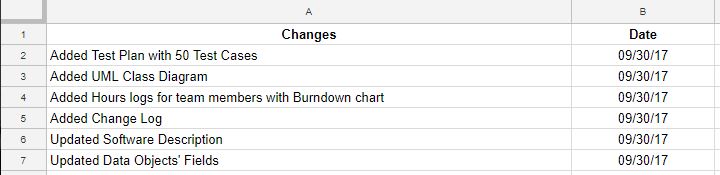
**vii. Milestones………………………………………………………………………………………….7**

**viii. Test Plan/Test Cases…………………………………………………………………………….8**

**ix. UML Class Diagram……………………………………………………………………………….9**

**x. Team Members Hours Log……………………………………………………………………….11**

**Change Log**



**Project Proposal**

**Team Name**

Team No Name Yet or Team No Name

**Members**

* Eric Ray - [ejray21@gmail.com](mailto:ejray21@gmail.com) (Programmer, Tester)
* Ervin Rodriguez - [mtr746@my.utsa.edu](mailto:mtr746@my.utsa.edu) (Lead, GUI Programmer, Design)
* Miguel Cardenas - [c.miguel709@gmail.com](mailto:c.miguel709@gmail.com) (Lead, Chief Architect, Design)
* Richard Salcedo - [kbgagt@gmail.com](mailto:kbgagt@gmail.com) (Programmer, Tester)
* Stefan Marchand - [marchandstefan@gmail.com](mailto:marchandstefan@gmail.com) (Programmer, Tester)

**Software Description**

Door Production Manager is an inventory management and blueprint creator application. It is designed to minimize the time needed to complete an order for a door and make the process easier for the user. It will minimize the hassle of manually keeping track of inventory and the process of creating a report. It allows users/company to upload sketches, pictures, or blueprints of the product that is easier to communicate with other users in the company or with customers.

First, a customer contacts a user/company to buy a door. The user checks the inventory and creates a list that will produce an estimated cost. Using the product information given by the customer and/or drafted by the user, the application generates a sketch or blueprint of the product. Based on the blueprint, the program will create a list of the required parts and materials. Then, the inventory is updated. When the product is completed, creates packing slip

**Feature List**

1. User Management; create and add users, manage levels of access.
2. Inventory SCRUD.
3. Order SCRUD.
4. Generate/Produce Reports: Inventory, Estimate or Quote, Blueprint.
5. Auto-generate a simple blueprint based on the measurements found in the Product.

**Target Platform**

This will be a Desktop Application.

**Alternate/Competing Products**

* **ABC Inventory** - Inventory Management application. Allows users to create purchase/work orders, keep track of inventory, and produce reports.
* **SmartDraw** - Blueprint maker application. Allows the creation of professional quality diagrams with the use of context specific templates and objects.

**Client Stakeholder**

Mr. Juan Cardenas (Deansteel)

**Glossary**

**SCRUD** - stands for Search, Create, Read, Update, Delete; general actions that a program performs when working with databases

**Inventory** - can also be referred to parts, these are parts that are used to build doors; items that are not sold directly to client

**Quote** - a list containing parts, materials, or product and a price estimate

**Order** - defines a transaction that was made or is being completed for a customer; contains information about the product and customer

**Product/Door** - Main product being sold by the company; items that can be sold

**Materials** - Anything used in the building process that is managed in the company’s inventory (sheet metal, wood, etc.)

**Hardware/Additions** - Additional features that can be sold with a door (hinges, locks, closing devices, windows, etc.)

**Blueprint** - Diagram detailing the various dimensions/measurements of a door

**User** - Company worker using the application (Salesman, Accountant, Stock Manager, etc.)

**Customer** - Individual who contacts the company asking for a quote on specific doors

**Inventory Database** - A database that contains Stock Products, Materials, and any Hardware/Additions needed to create all types of quotes

**Order Database** - A database that contains all Orders on record, past and recent

**Report** - Various documents that can be printed out for use within the company (required materials list, invoice, blueprints, etc.)

**Functional Requirements**

**User Login/Logout.** As a user, I want to be able to be able to login into or logout of the application.

**Create/Add User.** As an admin user, I want to be able to add other employees as user into this application.

**Remove User**. As an admin user, I want to be able to remove or delete inactive users from the application.

**Change Level of Access**. As an admin user, I want to be able to change a user’s level of access by defining the user type.

**Add Inventory Item**. As a user, I want to be able to add a possibly new item into the Inventory.

**Delete Inventory Item**. As a user, I want to be able to delete an item from the Inventory.

**Read Inventory Item**. As a user, I want to be able to view an item to check its properties.

**Search Inventory Item.** As a user, I want to be able to search for an item or a set of items.

**Update Inventory Item.** As a user, I want to be able to make changes or update an existing item in the Inventory.

**Add/Create Order**. As a user, I want to be able to start or create an order after a request from a customer.

**Delete Order.** As a user, I want to be able to delete or remove an existing order.

**Search Order**. As a user, I want to be able to search for a past or current order.

**Update Order**. As a user, I want to be able to make changes or update a current or pending order.

**Read Order**. As a user, I want to be able to view an order to read detailed information about that order.

**Add to Quote**. As a user, I want to be able to add an Inventory item into a Quote by selecting it.

**Save Quote**. As a user, I want to be able to save a Quote that is in process for a later use or reference.

**Upload Drawing/Blueprint**. As a user, I want to be able to upload an image of a blueprint or sketch into the application or specifically into an order.

**Delete Drawing/Blueprint**. As a user, I want to be able to delete a sketch from an existing order when it is not needed or contains incorrect informations.

**Add Notes to Drawing/Blueprint.** As a user, I want to be able to add notes or hand notes onto an uploaded drawing or blueprint.

**Share/Print Reports**. As a user, I want to be able to easily share the reports either by exporting it into a printable document or later on by email.

**User Types**

**All Users**

Login/Logout

Share/Print Reports

**Salesman**

Read Inventory Item

Search Inventory Item

SCRUD Quote

**Accountant**

SCRUD Order

Search Quote

Read Quote

Update Quote

**Stocker**

SCRUD Inventory Item

**Engineer**

Search Inventory Item

Read/View Inventory Item

Update Inventory Item

Search Order

Read Order

Update Order

Upload Drawing/Blueprint

Delete Drawing/Blueprint

Add Notes to Drawing/Blueprint

**Admin**

Create/Add User

Remove User

Change Level of Access

**Data Objects**

**USER**

* **Id (int):** the id of this user, used mainly in the database
* **User type/access (String):** the type of user or the level of access; Salesman, Accountant, Stocker, Engineer
* **Username/login (String):** the username used to login
* **Password (String):** the password used to login
* **Name (String):** the name of the user; used as a display
* **Email address (String):** email address of user; used when sharing orders or quotes

**INVENTORY**

* **Id (int)**: unique id for this part; used for database
* **Item number (String)** - a number assigned by Deansteel to this time
* **Quantity** **(int)**- the quantity of this part still available in the warehouse
* **Minimum quantity (int)**- the minimum quantity of this part that should be in the warehouse
* **Maximum quantity (int)**- the maximum quantity of this part in the warehouse
* **Manufacturer (String)**- the manufacturer of this part
* **Manufacturer Number (String)**- the manufacturer assigned number of this part
* **Vendors (String, comma-separated values)**- the vendors that are supplying this part
* **Size** **(String)**- the size description of this item
* **Color code** **(String)**- a color code for this item
* **Unit of measure** **(String)**- the unit of measure used for this item
* **Extra information (String)**- a String description of this item
* **Category (String)**- The category this item belong to; i.e. part, material
* **Accounting code** **(String)**- a code used by the accountant
* **Actual cost** **(double)**- the actual cost to obtain this item
* **Selling price** **(double)**- the selling price of this item
* **Taxable** **(boolean, String in DB)**- whether this item is taxable or not

**ORDER**

* **id (int)**: unique id for this order; used for DB
* **quote(Quote object), quoteID in db (int)**: the Quote object used for this Order; stored as quote ID in the DB
* **customerPurchaseOrderNumber (String)**: a number that is given by the customer
* **customerName (String):** the name of the customer this order is made for
* **productCode (String CSV)**: a unique product code provided by Deansteel, there can be 1 or multiple product codes within an order
* **status (String)**: the status of this product, automatically gets updated based on shipping dates
* **dateOrdered (Date)**: the date the Quote was approved
* **targetShipping** **(Date)**: taken from the Quote
* **actualShipping** **(Date)**: inputted later on before production
* **blueprint (Blueprint object), blueprintID in DB (int)**: Blueprint object that is used for this order; stored as a blueprint ID in DB
* **totalAmount (double)**: the total amount in this order

**PRODUCT**

- **productID:** unique number to identify product; for database

- **inventoryList (List<Inventory>), inventoryIDs CSV format in DB (String):** List of inventory items used to make this product; stored as a String comma-separated ID values in DB

**- totalCost** **(double)**: the total cost of the product based on the selling price of the inventory items used to make it

**QUOTE**

* **quoteID (int):** the id of this quote; for DB
* **productsList (List<Product>), productIDs CSV format in DB(String):** List of products used for this quote; stored as a String comma-separated ID values in DB
* **totalCost (double):** the total estimated cost for this quote, may include additional cost such as labor, etc...

**PRODUCT/QUOTE TABLE in DATABASE**

Product and Quote will share the Product Table in the Database since both have an id, a List, and a totalCost. In the database. There will be a 4th field called **category** to distinguish whether a record is either a **Product or Quote**.

**BLUEPRINT**

* **id (int):** unique identifying number; for database
* **productID (String)**: the id of the product this blueprint is made for
* **pdf/Image of blueprint:** for visual inspection

**REPORT (still need to research)**

* **content (String):** the content of this report; this will be generated by a toString() like function built in the Data objects to be printed
* **blueprints (List<Image>):** The blueprints in this report

**Milestones**

**Sprint 1**

1. Finish Specification
2. Have a GUI mockup
3. Set up a Version Control such as Github, BitBucket, or Unity VC

**Sprint 2**

1. Create GUI skeleton
2. Finish Inventory SCRUD
3. Finish Order SCRUD
4. Create GUI for inventory and order

**Sprint 3**

1. Finish Quote creation
2. Finish Reports creation
3. Product SCRUD
4. Create GUI for quote and reports

**Sprint 4**

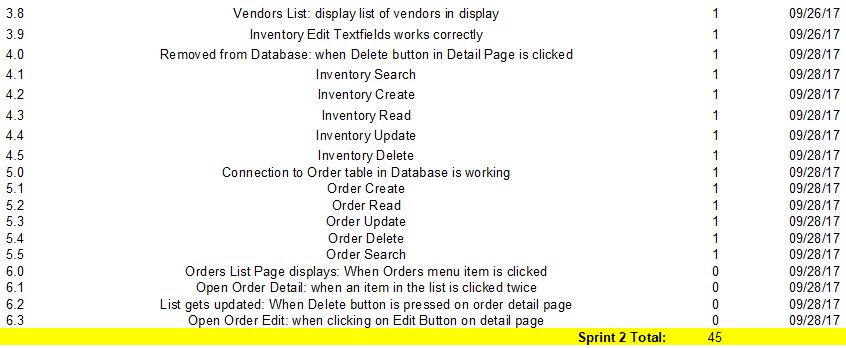
1. User Restriction
2. Finish Blueprint Uploading
3. Create GUI for blueprint

**Sprint 5**

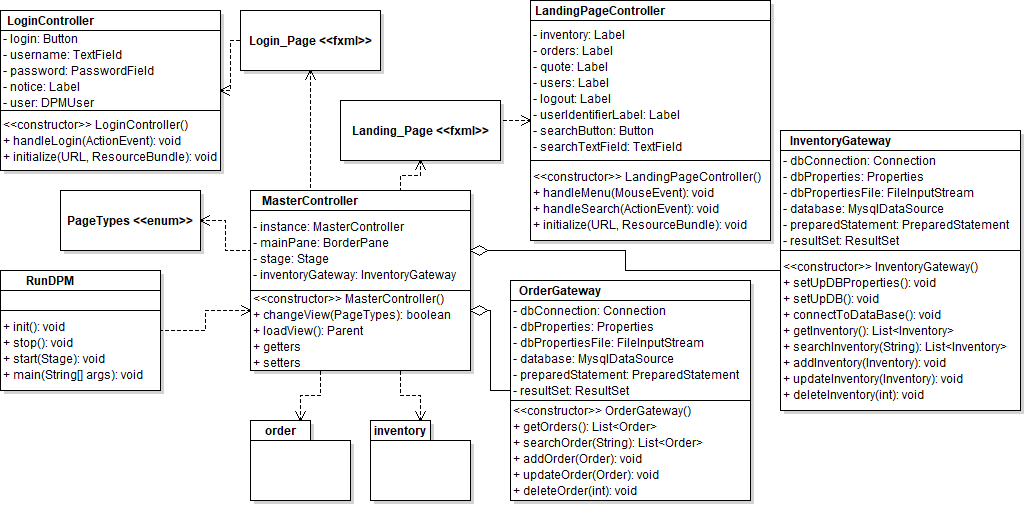
1. Implement additional features/optional features (login/user types)
2. Testing

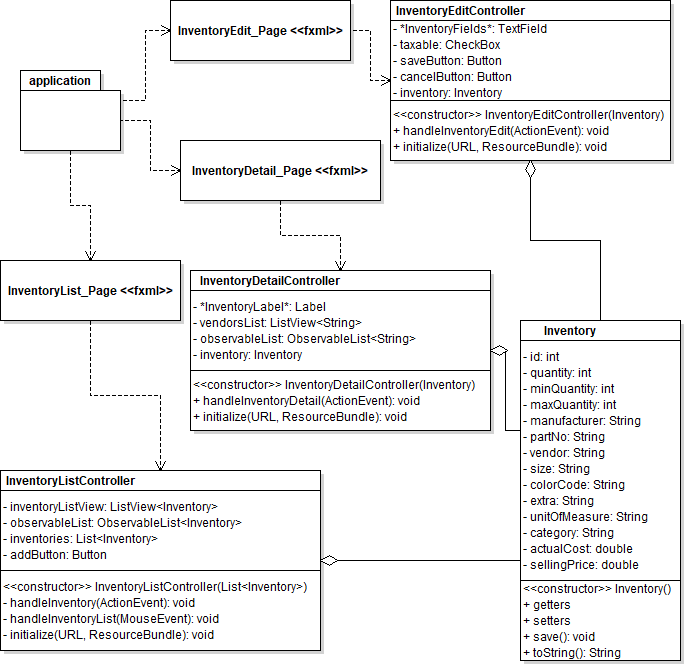
**Test Plan/Test Cases**

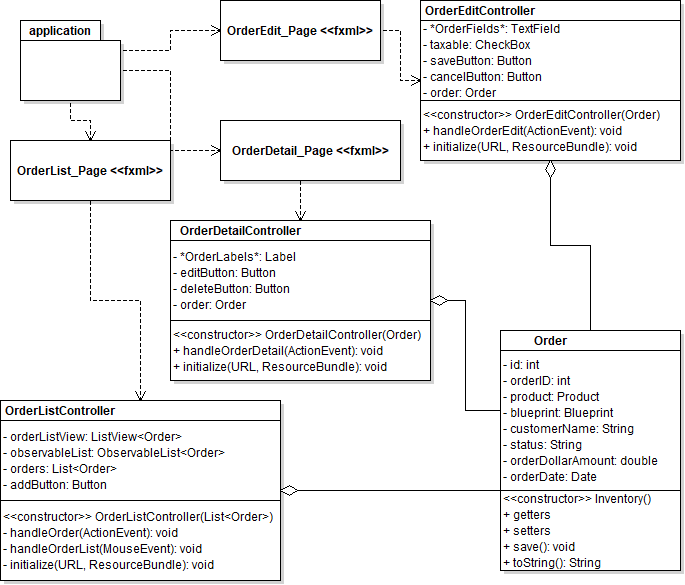




**UML Class Diagram**

****

****

****

**Team Member Hours Completed and Attempted**