

# **SOFTWARE DESIGN DOCUMENT**

Revision 1.0

September 7th, 2012

Prototype of Metalform Material Tracking System (PMMTS)

Minh N. HA

## Table of Contents

REVISION.....	4
<b>1. INTRODUCTION.....</b>	<b>5</b>
1.1. Document Purpose.....	5
1.2. Scope.....	5
1.3. Overview of Document.....	5
1.4 Prototype Requirements.....	5
1.5. Glossary.....	7
<b>2. DESIGN OVERVIEW.....</b>	<b>8</b>
2.1 Ruby on Rails (RoR) – Heroku.....	8
2.2 Deployment.....	8
2.3. Interface Design.....	8
2.4 Rejected Solutions.....	8
<b>3. ARCHITECTURE DESIGN.....</b>	<b>9</b>
3.1 APPLICATION ARCHITECTURE.....	9
3.2. AUTHENTICATION (LOG IN).....	9
Description .....	9
3.3. ADMIN FEATURES.....	10
Description.....	10
New Material Type.....	10
New Material.....	10
Report.....	10
Log Out.....	11
3.4. USER.....	11
Description.....	11
Choose off-cut.....	11
Update off-cut.....	11
Log Out.....	11
3.5. UPDATE LOCATION.....	12
Description.....	12
Check for all queued jobs.....	12
Update the location of the off-cut.....	12
Log Out.....	12
<b>4. PHASE I – Authentication (Log in).....</b>	<b>13</b>
4.1. Use Case Realisations.....	13
4.2. Data Structure Design.....	13
<b>5. PHASE II – User (Search and select off-cut).....</b>	<b>13</b>
5.1. Use Case Realisations.....	13
5.2. Data Structure Design.....	13
<b>6. PHASE III – User (Update off-cut).....</b>	<b>13</b>
6.1. Use Case Realisations.....	13

6.2. Data Structure Design.....	13
<b>7. PHASE IV – Update location.....</b>	<b>14</b>
7.1. Use Case Realisations.....	14
7.2. Data Structure Design.....	14
<b>8. PHASE V – Admin features.....</b>	<b>14</b>
8.1. Use Case Realisations.....	14
8.2. Data Structure Design.....	14
<b>REFERENCES.....</b>	<b>15</b>

**Illustration Index**

Illustration 1: Architecture Design..... 8

**Index of Tables**

Table 1: Glossary..... 6

Table 2: Username & Password..... 13

**REVISION**

Rev 1.0: document initiation

# 1. INTRODUCTION

## 1.1. Document Purpose

This Software Design Document (SDD) provides the design overview and details of the Prototype of Metalform Material Tracking System (PMMTS).

The expected audience of PMMTS is the technicians (x3) of the programming division of Metalform.

## 1.2. Scope

As the team lacks the technical expertise for the chosen development framework, the approach of the project is “learn as we go”. Each revision of the SDD will provide documentation up to the current development phase. The subsequent phase will be treated as “to be decided” (tbd.)

The reasons of choosing a development framework which the team does not have thorough experience will be explained in section 2.

By the end of the project this document will contain a complete description of the design of PMMTS.

The basic architecture is a web application built on top of Ruby on Rails (RoR) framework and PostgreSQL database. The basic pages will be in HTML and embedded Ruby. The prototype will be hosted on Heroku.

## 1.3. Overview of Document

Section 2:

- The reasons why RoR was chosen.
- The location for hosting the prototype.

Section 3:

- The Architectural Design that specifies the design entities that collaborate to perform all the functions included in the system.
- Each of these entities has an Abstract description concerning the services that it provides to the rest of the system. In turn, each design entity is expanded into a set of lower-level design operations that collaborate to perform its services.

Section 4, 5, 6, 7, 8:

- The detailed design of each entity.

## 1.4 Prototype Requirements

- One (1) dedicated tester.
- At least three (3) dedicated racks for testing purpose.

- Please suggest a naming standard for them.
- Allocate several types of material of various thicknesses for testing purpose.
  - Example:
    - Electrogalv Steel (EG): 0.6, 3mm.
    - Galvanised Steel (Galv) 0.55, 3mm.
    - Corten (350HW) 3, 6mm.
  - The ID of the off-cut can only be allocated manually at this state.
    - Suggested ID naming standard: [Material code] + [thickness in mm] + [-ID]
    - Example: EG060-001, GALV055-001, 350HW060-001 etc.
- Please provide us with the list of materials so that we can populate the test database with real data.

## 1.5. Glossary

SDD	Software Design Document
Tbd.	To be decided
PMMTS	Prototype of Metalform Material Tracking System
Open-source	Indicate, but not limit to, a program whose source code is openly and freely available
Ruby	An open-source high level programming language <a href="http://www.ruby-lang.org/en/">http://www.ruby-lang.org/en/</a>
Embedded Ruby	Ruby code embedded within a web page
Ruby on Rails (RoR)	An open-source web development framework which features the Ruby programming language <a href="http://rubyonrails.org/">http://rubyonrails.org/</a>
Rails	Short for Ruby on Rails
C#	A proprietary programming language from Microsoft
ASP.NET	A proprietary web application framework from Microsoft
RDBMS	Relational Database Management System
PostgreSQL	An open-source RDBMS <a href="http://www.postgresql.org/">http://www.postgresql.org/</a>
Postgres	Short for PostgreSQL
MSSQL	Microsoft SQL Server. A proprietary RDBMS
MySQL	An open-source RDBMS from Oracle <a href="http://www.mysql.com/">http://www.mysql.com/</a>
PaaS	Platform as a service
Heroku	A cloud based application platform service (PaaS) <a href="http://www.heroku.com/">http://www.heroku.com/</a>
HTML	Hyper Text Markup Language
Off-cut	A particular instance of material

Table 1: Glossary

## 2. DESIGN OVERVIEW

### 2.1 Ruby on Rails (RoR) – Heroku

A remote hosted solution was chosen because the team needs feedbacks from the user in order to further and refine the design.

Rails<sup>i</sup> is a full-stack open-source framework, meaning that it gives the web developer the ability to gather information from the web server, talk to or query the database, and render templates.

Heroku<sup>ii</sup> is a cloud based application platform service (PaaS) which supports many programming language including Ruby. Heroku provides the service free of charge for small application deployments. Upgraded plans can be purchased if more processing power or sophisticated management is required. The main database backed for a Heroku deployment is PostgreSQL.

PostgreSQL<sup>iii</sup>, or simply Postgres, is an open-source object-relational database management system (RDBMS). It is ACID-compliant<sup>iv</sup> (atomicity, consistency, isolation, durability) therefore is well suited for a production system.

By utilising Rails and Heroku, the prototyping process can be securely and efficiently provided.

### 2.2 Deployment

<http://pmmts.herokuapp.com/>

### 2.3. Interface Design

The interface will be the basic HTML forms generated by Rails. It will not be thoroughly designed and styled.

### 2.4 Rejected Solutions

Due to the constraints of distance and the time a locally hosted application will not be suited for the active prototyping development process which is a necessity of this project. The locally hosted applications include:

- Windows form written in C# and using Microsoft SQL Server (MSSQL) as database backend
- Windows form written in C# and using MySQL as database backend
- Locally hosted web application written in ASP.NET using MSSQL database backend
- Locally hosted web application written in ASP.NET using MySQL database backend

An affordable and reliable hosting provider for ASP.NET application was not identifiable therefore the following solution was also rejected:

- Remote hosted web application written in ASP.NET using MySQL database backend.



### 3. ARCHITECTURE DESIGN

#### 3.1 APPLICATION ARCHITECTURE

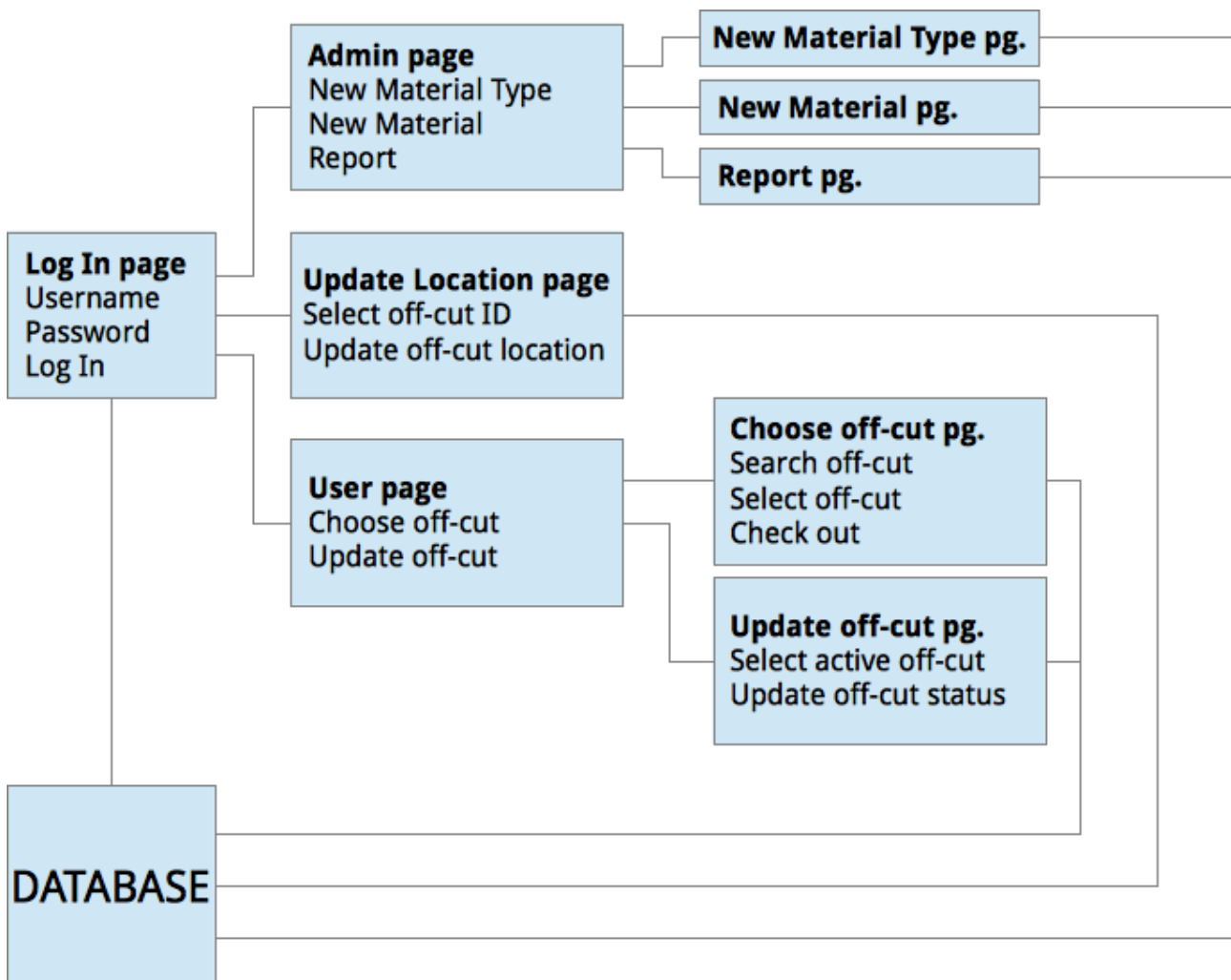


Illustration 1: Architecture Design

#### 3.2. AUTHENTICATION (LOG IN)

##### Description

The user will be presented with a Log In page which have two textboxes “Username”, and “Password” and one button “Log In”. The user needs to type in his/her “Username” and “Password” and then click on the “Log In” button. If the username/password combination is correct, the user will be taken to the subsequent pages according to the privileges of the username.

### 3.3. ADMIN FEATURES

#### Description

The user will be presented with the “Admin page” which has four buttons: “New Material Type”, “New Material”, “Report”, and “Log Out”. Clicking on each of the button will take the user to the associated page.

#### New Material Type

##### Add a new type of material into the database

- Choose the type of material from a drop down list.
- If the intended type is not in the database, the user can enter a new one.
- **Example:** existing material types: Electrogalv Steel (EG): 0.6, 3mm (EG060, EG030).
  - To be added: Electrogalv Steel (EG): 5mm.
    - Choose EG from the drop down list.
    - Type in thicknesses: 050.
    - New type of material will be created: EG050
  - To be added: Galvanised Steel (Galv) 0.55mm
    - Type in type: GALV
    - Type in thicknesses: 055.
    - New type of material will be created: GALV055

#### New Material

##### Add a new instance of material into the database

- Choose the type of material from a drop down list.
- Choose the thickness from a drop down list.
- Manually enter the ID for the off-cut.
- Enter the measurements of the off-cut (height & width)
- The system will display the latest ID of the chosen material. The ID of the new off-cut will increase by one
- **Example:** to be added Electrogalv Steel 0.6mm
  - Choose Electrogalv Steel from the drop down list.
  - Choose the thicknesses of 0.6mm.
  - The system indicated that the latest ID of Electrogalv Steel 0.6mm is EG060-002.
  - Type in 003 to the ID field. The new ID will be EG060-003.

#### Report

##### Check for all available off-cuts which are still stored on racks

- This is for accounting, stock take purpose

- At this state, the system can only display all available off-cuts on a web page
- The user must manually enter these numbers into the accounting software.

#### **History of off-cuts**

- Check what job(s) have been allocated to which off-cut.

### **Log Out**

The current user will lose the admin privileges and will be re-directed to the “Log In page”

## **3.4. USER**

### **Description**

The user will be presented with three buttons: “Choose off-cut”, “Update off-cut”, and “Log out”. Clicking on each of the button will take the user to the associated page.

#### **Choose off-cut**

- Check the database for:
  - Material type
  - Thickness
- The system will list all suitable off-cuts
- If a suitable off-cut is found:
  - The user will enter the job ID which is provided by the accounting software.
  - Check that off-cut in order to program it (the off-cut is temporarily taken out of the database).

#### **Update off-cut**

- The user will select the job ID which is still being checked out.
- Update the status of the off-cut:
  - Used up.
  - Not used up:
    - Enter the new measurements (height & width)
- Finalising the update, the following will be done automatically by the systems:
  - Flag the off-cut as used up in the database.
  - Place the off-cut back to the database with new measurements if it is not used up.
  - Place this particular job ID in to a queue.

### **Log Out**

- The current user will lose the privileges and will be re-directed to the “Log In page”

### 3.5. UPDATE LOCATION

#### Description

The user will be able to update the location of off-cut on the racks and on the workspace.  
Detail description is tbd.

#### Check for all queued jobs

- The system will:
  - Display all queued job
  - Display the location of the off-cuts required

#### Update the location of the off-cut

- **Remove:** if an off-cut is taken out the rack
- **Return:** if an off-cut is placed back to the rack

#### Log Out

- The current user will lose the privileges and will be re-directed to the “Log In page”

## 4. PHASE I – Authentication (Log in)

### 4.1. Use Case Realisations

The user who types in “admin” will be directed to “Admin page”

The user who types in “user” will be directed to “User page”

The user who types in “forklift” will be directed to “Update Location page”

### 4.2. Data Structure Design

The “username” and “password” will be populated by the team as at this state

Username	Password	Description
admin	admin	Admin user, can access “Admin Features”
tester1	tester1	The test user, can access “User Features”
forklift	forklift	The user who will operate the forklift to change the location of off-cuts, can <u>only</u> access “Update Location”

Table 2: Username & Password

## 5. PHASE II – User (Search and select off-cut)

### 5.1. Use Case Realisations

Tbd.

### 5.2. Data Structure Design

Tbd.

## 6. PHASE III – User (Update off-cut)

### 6.1. Use Case Realisations

Tbd.

### 6.2. Data Structure Design

Tbd.

## **7. PHASE IV – Update location**

### **7.1. Use Case Realisations**

Tbd.

### **7.2. Data Structure Design**

Tbd.

## **8. PHASE V – Admin features**

### **8.1. Use Case Realisations**

Tbd.

### **8.2. Data Structure Design**

Tbd.

## REFERENCES

Wikipedia contributors. (2012a, September 5). Ruby on Rails. *Wikipedia, the free encyclopedia*. Wikimedia Foundation, Inc. Retrieved from [http://en.wikipedia.org/w/index.php?title=Ruby\\_on\\_Rails&oldid=510870261](http://en.wikipedia.org/w/index.php?title=Ruby_on_Rails&oldid=510870261)

Wikipedia contributors. (2012b, September 5). Heroku. *Wikipedia, the free encyclopedia*. Wikimedia Foundation, Inc. Retrieved from <http://en.wikipedia.org/w/index.php?title=Heroku&oldid=508977205>

Wikipedia contributors. (2012c, September 5). PostgreSQL. *Wikipedia, the free encyclopedia*. Wikimedia Foundation, Inc. Retrieved from <http://en.wikipedia.org/w/index.php?title=PostgreSQL&oldid=510746064>

Wikipedia contributors. (2012d, September 5). ACID. *Wikipedia, the free encyclopedia*. Wikimedia Foundation, Inc. Retrieved from <http://en.wikipedia.org/w/index.php?title=ACID&oldid=510479231>

- i Ruby on Rails (Wikipedia contributors, 2012a)
- ii Heroku (Wikipedia contributors, 2012b)
- iii PostgreSQL (Wikipedia contributors, 2012c)
- iv ACID (Wikipedia contributors, 2012d)