Masters Project Final Report

December 2015

Project Title	Production Tracking System for Brandix Casualwear Seeduwa			
Student Name	H N D G A Weerakoon			
Registration No. & Index No.	2013/MIT/089 13550899			
Supervisor's Name	Mr. G. K. A. Dias			
Please Circle the appropriate	**Master's Program**		**Type**	
	MIT	**~~MCS~~**	**~~Research~~**	**Implementation**

For Office Use ONLY

Production Tracking System for Brandix Casualwear Seeduwa

H N D G A WEERAKOON

2016

Production Tracking System for Brandix Casualwear Seeduwa

A dissertation submitted for the Degree of Master of Information Technology

H N D G A WEERAKOON

University of Colombo School of Computing

2016

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: H N D G A Weerakoon

Registration Number: 2013/MIT/089

Index Number: 13550899

____

Signature Date: 28/12/2015

This is to certify that this thesis is based on the work of

Mr. H N D G A Weerakoon

Under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Mr. G. K. A. Dias

____

Signature Date: 28/12/2015

Abstract

This Production Tracking System was developed for Brandix Casualwear Seeduwa (Pvt) Limited. This system is for the benefit of the employees who are directly involved in the factory production line and the factory management.

The current way of Production tracking in the factory is fully manual. In the current process the Brandix Seeduwa factory is marking Budgeted and actual (Hourly) production details in excel files and when management want to check details they have to go through excel files and get the details. Actual details are marked every hour in a White board near the production lines in the factory where the management can’t get the exact details unless they visit and check the details. Due to this process management have issues where they are unable to take immediate actions if the line production is not up to the expected numbers. And management is not aware about the total monthly output with current day’s actual details until the end of each day.

The purpose of this project will be to overcome all the above mentioned issues and to build a system which will help the Factory to work more easily with their day to day activities. “Production Tracking System” will be a solution where all the actual details will be available and stored. And it will contain the budgets of each factory line and all these information will be stored in the Data Entry module side of the system. With all those details management will be able to view hourly and the details of the lines and they have the possibility of comparing those with the budgeted values and take necessary decisions to make the overall production output more efficient and all these can be done using the Dashboard module of the system. Also management is able to see any feedback from the line immediately in the dashboard where the communication between the line staff and the management will be faster. Each hour with the latest Hourly production numbers of each line displaying against the budget values of the lines will be a value addition to the management stuff where to get all details. Also management have the ability to add new lines when the factory grow.

Prototype methodology was used for the implementation of the project and production Tracking System is developed using C# and ASP.NET programing languages. Kendo UI framework is used as the standard charting options to the system.

This system facilitates the objective of delivering a simple and versatile solution to production tracking details of each factory line.

Acknowledgments

Author's heartfelt gratitude goes to his Final Year Project Supervisor Mr. G. K. A. Dias for providing me with all the fundamentals and all the support given thought out this project. Without his assistance this project would have been impossible.

I take this opportunity to express my thanks to staff of University of Colombo School of Computing, for all the encouragements given towards the success of this assignment.

Last but not least I would like to thank my family who was helping me to finish this project and friends who helped in many ways to make this project a successful one.

Table of Contents

Declaration....ii

Abstract....iii

Acknowledgments....iv

Chapter 1: Introduction....1

1.1 Motivation....1

1.2 Objective....2

1.3 Scope of the Project....2

1.4 Proposed Solution....2

1.5 Benefits to the Organization....3

1.6 Outline of Report....3

Chapter 2: Background....4

2.1 Literature Review....4

2.1.1 Tableau....8

2.1.2 G.PRO Shop floor Data Tracking....8

2.1.3 Pros and cons....9

Chapter 3: Analysis and Design....10

3.1 Analysis....10

3.1.1 Requirement Gathering Methods....10

3.1.2 Functional Requirement....10

3.1.3 Non-Functional Requirement....11

3.2 Design....12

3.2.1 Development Methodology....12

3.2.2 Use Case Diagram....13

3.2.3 User Characteristics....19

3.2.4 Principle Actors....19

3.2.3 Activity Diagram....20

3.2.5 Sequence Diagram....21

3.2.4 ER Diagram....23

Chapter 4: Implementation....24

4.1 Introduction....24

4.2 Technical Background....24

4.3 Development Tools 24

4.4 Database Scripts 25

4.4 Module Structure of the System 26

4.4.1 Business Layer 26

4.4.2 Display Layer 26

4.4.3 Input Control 27

Chapter 5: Evaluation 28

5.1 Test Process 28

5.1.1 White Box Testing 28

5.1.2 Black Box Testing 29

5.2 Testing Levels 29

5.2.1 Unit Testing 29

5.2.2 Integration Testing 29

5.2.3 Validation Testing 29

5.2.4 System Testing 30

5.3 Test Cases 30

5.4 Test Data 33

5.5 User Evaluation 34

Chapter 6: Conclusion and Future Work 37

6.1 Introduction 37

6.2 Results with Respect to the Objectives 37

6.3 Critical Appraisal 37

6.4 Future Work 37

Appendices 40

Appendix A: Wire frames 40

Appendix B: Database Scripts 44

Appendix C: Test Cases 78

List of Figures

Figure 1 : Brandix Group of Companies (Source: Brandix.com [1]) 1
Figure 2 : Bundle Number 5
Figure 3 : SOBAR sheet 5
Figure 4 : Stickers 7
Figure 5 : Tableau sample dashboard (Source: Tableau.com [2]) 8
Figure 6 : SDT Sample chart and tags (Source: g. technologies [3]) 9
Figure 7 : Development Methodology 12
Figure 8 : Use case 13
Figure 9 : User registration Activity diagram 20
Figure 10 : Admin and Dashboard Sequence diagram 21
Figure 11 : Admin and add lines Sequence diagram 22
Figure 12 : Data entry user and Add budget Sequence diagram 22
Figure 13 : ER diagram 23
Figure 14 : MVC model diagram 26
Figure 15 40
Figure 16 40
Figure 17 41
Figure 18 41
Figure 19 42
Figure 20 42
Figure 21 43
Figure 22 43

List of Tables

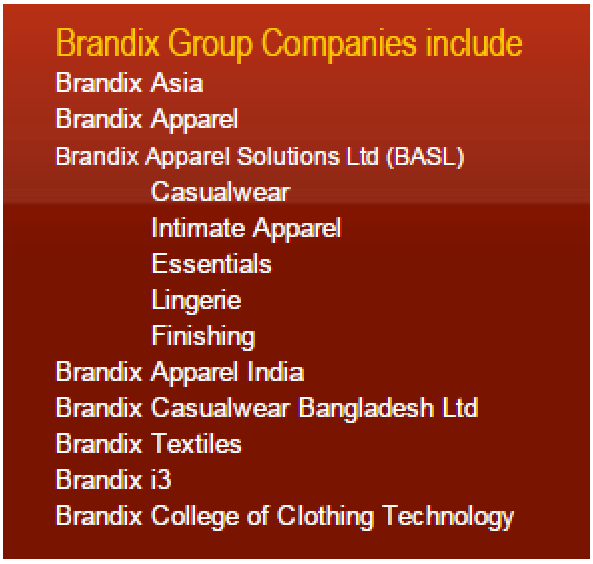
Table 1 : Pros and Cons analysis 9
Table 2 : Non Functional Requirement 11
Table 3 : Register use case 14
Table 4 : Add budget use case 14
Table 5 : Edit Budget use case 15
Table 6 : View Dashboard use case 15
Table 7 : Add actuals use case 16
Table 8 : Edit Actuals use case 17
Table 9 : Add Lines use case 17
Table 10 : Edit Lines use case 17
Table 11 : View lines use case 18
Table 12 : Edit master data use case 18
Table 13 : Login use case 19
Table 14 : Test Cases 32
Table 15 : UAT Test case 36

List of Abbreviations

MIT	**-**	**Masters of Information Technology**
IT	**-**	**Information Technology**
BCS	**-**	**Brandix Casualwear Seeduwa**
PTS	**-**	**Production Tracking System**
KPI	**-**	**Key Performance Indicator**
ERP	**-**	**Enterprise Resource Planning**
MVC	**-**	**Module-View-Controller**
UAT	**-**	**User Acceptance Testing**

Chapter 1: Introduction

“Brandix is largest end-to-end apparel exporter in the Sri Lanka” [1]. Brandix group includes more than 10 core companies where 40 plus factories are under those companies.



Brandix Group Companies include

- Brandix Asia
- Brandix Apparel
- Brandix Apparel Solutions Ltd (BASL)
  - Casualwear
  - Intimate Apparel
  - Essentials
  - Lingerie
  - Finishing
- Brandix Apparel India
- Brandix Casualwear Bangladesh Ltd
- Brandix Textiles
- Brandix i3
- Brandix College of Clothing Technology

Figure 1 : Brandix Group of Companies (Source: Brandix.com [1])

Each of these factories have their own processes of creating the garment materials. Most of these factories run the ERP system named Lawson M3 to support the production process and tracking of the production. Brandix Casualwear Seeduwa (BCS) is one of the factories among the others which have a different process where there main garment materials are shorts and trousers. Since the process is different to other factories BCS don’t have a proper tracking system, and currently production tracking is done using excel files which takes and long time and effort to give a final number of the production details. Idea of this project is to make a system which will enable BCS staff to input all the details to one centralized place and move out from marking details in white boards and excel sheets, Along with that to help management to track the factory KPI’s easily in a dashboard.

1.1 Motivation

The current process the Brandix Seeduwa factory is to mark the Budgeted and Hourly details excel files and when needed go back and check for the details. Hourly details are marked every hour in a White board in the factory were the management can’t get the exact details unless they visit and check the details from the factory line. Due to this process management have issues were they are unable to take immediate actions if the line production is low. And management is not aware about the total monthly output with current day’s actual details until the end of each day.

1.2 Objective

The objective of this project is to create a configurable Production Tracking System which can be adjusted based on the factory line count. And a system which can take in planned details of each factory line, along with hourly production input details. And a Management Dashboard which includes Key performance Indicators of the factory, which will be updated hourly based on the details available in the Production Tracking System.

1.3 Scope of the Project

The Scope of the project can be divided following points,

- Admin should be able to configure master data at the beginning of the project (Number of lines in the factory)
- User should be able to add monthly budget values
- User should be able to edit the monthly budget values
- User should be able to add hourly actual values
- Admin should be able to edit hourly actual values
- Admin should be able to add new users to the Tracking system & Management Dashboard
- Dashboard will contain 4 Key Performance Indicators
- Dashboard will show Key Performance Indicators to users based on the access level of the user

Proposed solution will be a customized Production tracking system which will help the management and staff of BCS to do their day today work more effective and efficient way.

1.4 Proposed Solution

The purpose of this project will be to overcome all the above mentioned issues and to build an system which will help the Factory to work more easily with their days to day activities. “Production Tracking System” will be a solution where all the Factory actual details will be available and stored. And it will contain the budgets of each factory lines and all these information will be stored in the Data Entry module side of the system. Which all those details Hourly management will be able to view the details of the lines and they have the possibility of comparing those with the budgeted values and take necessary decisions to make the overall production output more efficient and all these can be done using the Dashboard module of the system. Also management is able to see any feedback from the line immediately in the dashboard where the communication between the line staff and the management will be faster.

Each hour with the latest Hourly production numbers of each line displaying against the budget values of the lines will be a value addition to the management stuff where to get all details in the past they had to go through many documents.

1.5 Benefits to the Organization

- Make an efficient way of storing Production Hourly and Budgeted details
- Display information in a easy to understand manner
- Have the ability to configure new line when factory increase new lines

1.6 Outline of Report

Chapter 2 of the report will give an overview regarding the Production Tracking system in the industry. A literature review included with few applications which are with similar objectives and those will be critically analyzed.

Chapter 3 includes a detailed overview of Functional and Non – Functional requirements of the project. The UML and ER diagrams are included in the design phase of the same chapter.

Chapter 4 contains of the details regarding technical background and implementation of the project.

Chapter 5 is focused on the quality aspect of the project. This chapter includes the testing process and types considered and used. User evaluation and Sample test case also includes on this chapter.

Chapter 6 contains the conclusion of the project were Results with respect to the objectives, Critical appraisal and future work is discussed in this chapter.

Chapter 2: Background

In this chapter Literature review will contain the current production process of Brandix Casualwear Seeduwa and few other similar systems which are available will be discussed and analyzed.

2.1 Literature Review

In apparel Industry there are different type of scales, high scale Factories most of the time use top ERP systems support their production. SAP, Oracle, Infor , Microsoft Dynamics and IFS ERP systems are few among the top 10 ERPs in the world. Most of these top ERP’s have a high price range from $20K to $500K. In the Medium and Small scale factories most of the time use web based solutions and customized solutions to cater there production needs.

Also in apparel Industry the process of the Factory plays a big role, all the top class ERP are created towards few standard processers where for some factories it’s hard to adopt to the standard process. And some factories prefer doing there planning manually and track the production manually without using a system because of the process and the type of apparel garment which they produce.

In some small scale factories they use excel and white boards to track their production needs without using a production tracking system. In those factories Beginning of every month planning details are created in excel spreadsheets with the involvement of the management and those are saved in files. And the daily the hourly production of each factory line also tracked using white boards and excel spread sheets. End of each day line supervisors mark the actual production output. Later management look at all these details and compare planned vs Actual to create factory Key performance Indicators.

One of the issues in most of the available systems are that those expect the factories to change their current process of working. For top management and staff this is a big risk since the efficiency of the production can go down when the process is changed.

Because of mentioned reasons above most of the factories try to build systems aligned with their current process which will make more effective for their production.

The cut cloth bundles are taken to the beginning of the line where numbering of each and every item take place. An information sheet called SOBAR sheet (*Figure 3*) comes with the bundles.

A bundle can’t be divided to different lines, but the same SOBAR sheet can go to different lines (all bundle information except relevant are cut off).

Numbering is done according to the size of the garment and the bundle number (job number).

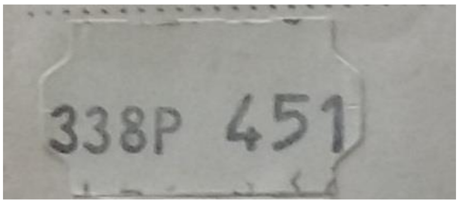
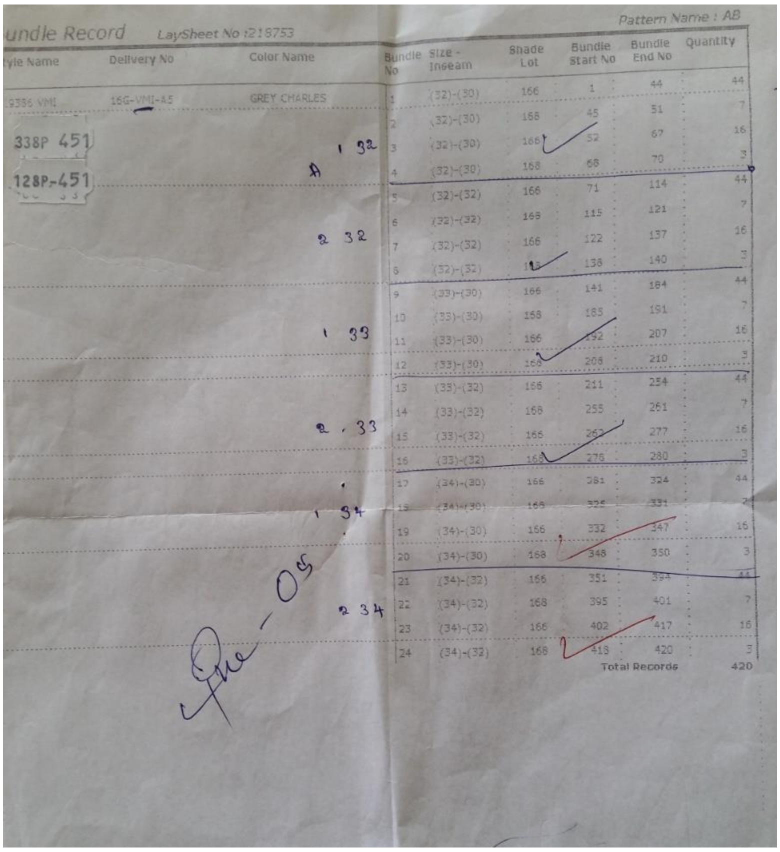


Figure 2 : Bundle Number



undle Record LaySheet No :218753

Pattern Name : AB

yle Name	Delivery No	Color Name	Bundle No	Size - Inseam	Shade Lot	Bundle Start No	Bundle End No	Quantity
9386 VMI	16G-VMI-A5	GREY CHARLES	1	(32)-(30)	166	1	44	44
			2	(32)-(30)	168	45	51	7
			3	(32)-(30)	166	52	67	16
			4	(32)-(30)	168	68	70	3
			5	(32)-(32)	166	71	114	44
			6	(32)-(32)	168	115	121	7
			7	(32)-(32)	166	122	137	16
			8	(32)-(32)	168	138	140	3
			9	(33)-(30)	166	141	184	44
			10	(33)-(30)	168	185	191	7
			11	(33)-(30)	166	192	207	16
			12	(33)-(30)	168	208	210	3
			13	(33)-(32)	166	211	254	44
			14	(33)-(32)	168	255	261	7
			15	(33)-(32)	166	262	277	16
			16	(33)-(32)	168	278	280	3
			17	(34)-(30)	166	281	324	44
			18	(34)-(30)	168	325	331	7
			19	(34)-(30)	166	332	347	16
			20	(34)-(30)	168	348	350	3
			21	(34)-(32)	166	351	394	44
			22	(34)-(32)	168	395	401	7
			23	(34)-(32)	166	402	417	16
			24	(34)-(32)	168	418	420	3
						Total Records		420

Figure 3 : SOBAR sheet

A single garment should be assembled with parts that are numbered the same. This is done to maintain consistency of color and size.

Currently there is no process to identify a missing number. It is checked manually by comparing the previous and the next pieces.

After the pieces are numbered, the bundles are proceeded to the line where sewing happens. Each machine operator will assemble each parts of the same number and progressively a garment will be completed at the end of the line.

- A single style will be completed within a line. In most cases the garment in completed within the line. But there may be situations where parts of the process is done in a separate line due to line balancing purposes. E.g.: The belt loops may be attached in a different line.
- Garment parts will be assembled according to the numbers given earlier.
- Every operator will take an hourly count of the pieces completed by them. Each operator notes it down on their notebooks.

A count is taken in the middle of the line where the completed front and back pieces are to be assembled. This is taken hourly and written on a board. There might be a brought forward balance at the beginning of the day (buffer). Hourly targets and faulty processes are checked through this.

All the labels, including the delivery number (cut number), line number, fabric care, style and color are attached.

- 1 cut number can represent only one shade of garment. But a shade of garment can be represented by different cut numbers depending on the orders and shipments.
- For cases where the order is a VMI (long term and high quantity orders where cut bundles are always kept in storage) a substitute cut number is given instead the delivery number.

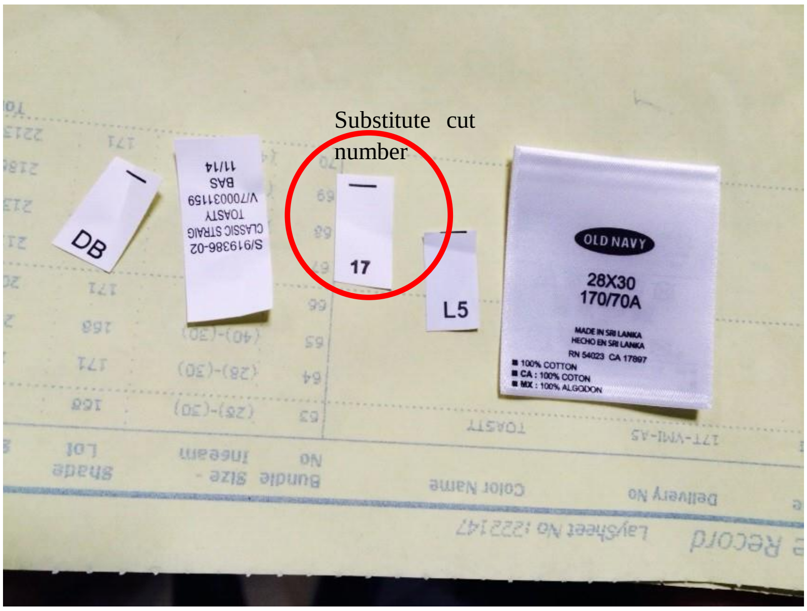


Figure 4 : Stickers

At the end of a sewing line, machine operators make sure that all the stickers are removed. This task is normally given to someone who has less work load. E.g.: The operator attaching the tack buttons

Finally, Quality checking is done to ensure that all the output garments are of right measurements and there are no damages. If any faulty item was found, the damaged area is marked and sent back to the line.

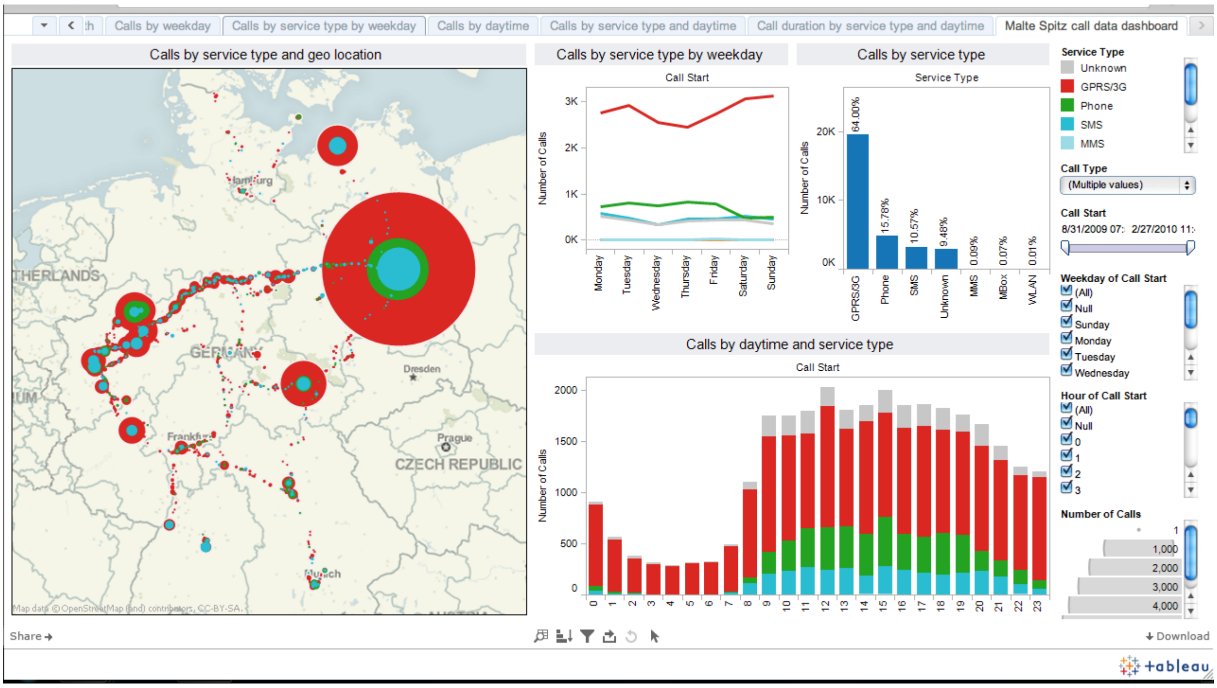
- Every item that passed the quality check is transferred to a bin as a batch and the quality checker will write down the number on a board. This count is not classified of the color/delivery number, it is just the total output of the line.
- A Line supervisor will take the hourly count of this and update a shared spreadsheet. This document is used only for internal efficiency monitoring.

Time to time, recorders will collect and classify the output items from the bin according to the cut/delivery numbers and send to washing. These numbers are written on a recording book.

The Line supervisor will get the counts from the above recording book and record it on a sheet called “Daily Output Report”. End of each day management will look at these actual numbers and compare against the budgeted numbers.

2.1.1 Tableau

Tableau [2] is dashboard software that allows non-technical users to create interactive, real-time dashboards in minutes. Adding filters and drilling down is as simple as a few clicks. Sharing a Tableau dashboard on Tableau Server or Tableau Online and embedding them into wikis, corporate portals or any web page is point-and-click. And visual best practices are baked in, so you can build a great dashboard and communicate effectively.



Call Type
(Multiple values)

Call Start
8/31/2009 07: 2/27/2010 11:

Weekday of Call Start
☑ (All)
☑ Null
☑ Sunday
☑ Monday
☑ Tuesday
☑ Wednesday

Hour of Call Start
☑ (All)
☑ Null
☑ 0
☑ 1
☑ 2
☑ 3

Figure 5 : Tableau sample dashboard (Source: Tableau.com [2])

Tableau is a dashboard software that allows users to create their own dashboard and adding filters and drilling down options to the dashboard. Tableau is a purely a dashboard making software were only have the option to make creative dashboards based on a given data source. Tableau don't allow users to create and maintain their own data source which is one of the key requirements for the Production tracking system.

2.1.2 G.PRO Shop floor Data Tracking

G.PRO Shop floor Data Tracking (SDT) [3] is a real time shop floor data collection and feedback system designed specifically to meet the production control and management needs of the apparel industry. This solution is created from GPRO Technologies in Malaysia.

G.PRO SDT uses RFID Smart Tags that has been integrated into the shop floor form cutting to finishing. Data is captured when events take place from the cutting floor to finishing. SDT has the ability to handle thousands of transactions per day and have the ability to give the output to a dashboard.

Figure 6 : SDT Sample chart and tags (Source: g. technologies [3])

G.PRO Shop floor Data Tracking contains most of the requirements of the customer of this project, but because this solution contain RFID tags it does not match to the current factory process. Also the cost to implement G.PRO Shop floor Data Tracking is very high.

2.1.3 Pros and cons

The pros and cons of each system analyzed above, are as follows;

System	**Pros**	**Cons**	**Why it's not suitable for Brandix Casualwear Seeduwa**
Tableau	It's a software which caters for dashboard users. Good chart visualization options	User need to have his own data source.	This system does not allow the users to maintain their data source.
G.PRO Shop floor Data Tracking	Support for the full process of the Apparel industry form Cutting to the finishing.	Support only RFID for the tracking process	This system requires a large implementation cost Only work with RFID tags, and to implement RFID the current process need to be changed.

Table 1 : Pros and Cons analysis

It was identified in the above analysis full solutions which cater to the requirement is very much costly and do implement that sometimes the current process needs to be changed. Also some other solutions which are really good only contain one aspect of the requirement and not the full requirement.

Chapter 3: Analysis and Design

3.1 Analysis

3.1.1 Requirement Gathering Methods

Following requirement gathering methods were used to collect requirements for this project,

- **Interviews**

To get the Functional Requirements of the project interviews were carries out with Factory management and Factory line Mangers and Production monitoring stuff. Factory Management explained how critical for them to see the hourly actual values to take immediate decisions. And Factory Production monitoring stuff explained how current process takes more time for them when it comes to reporting numbers.

- **Observations**

To understand the current business process and to study the full process observations were carries out. For Factory line production daily work and how management take decisions based on the daily details were specially monitored in this process. Clarifications which came up in the observation stage were later clarified with the Factory management to get the proper knowledge of the current process.

3.1.2 Functional Requirement

Functional Requirements contains the core requirements which are expected form the proposed solution. These explain the inputs and outputs of the solutions based on the customer expectations.

- Admin should be able to configure master data at the beginning of the project (Number of lines in the factory)
- User should be able to add monthly Planned values
- User should be able to edit the monthly planned values
- User should be able to add hourly actual values
- Admin should be able to edit hourly monthly values
- Admin should be able to add new users to the Tracking system & Management Dashboard
- Management Dashboard will contain 4 Key Performance Indicators

- Management Dashboard will show Key Performance Indicators to users based on the access level of the user

Proposed solution will be a customized Production tracking system which will help the management and staff of BCS to do their day today work more effective and efficient way.

3.1.3 Non-Functional Requirement

Non Functional requirements are the non-technical requirements of the project. But some of these requirements are essential for the system to operate.

NFR Category	NFR	Applicable
User Interface	Dashboard support for different devices?	Windows 7, 8 devices
	Support commonly used resolution sizes	Min 1024 x 768
	Loading Page loading time	5 – 8 Sec
	Individual KPI Chart loading time	2 – 3 Sec
Accessibility	Compliance to accessibility (Color blind users and etc.)	No
	System Accessible Minimum Network Bandwidth: Standard Location Bandwidth	Will be Accessible for the internal stuff of the factory since it will be hosted inside the factory.
Availability	Server failures and network downtimes will stop system availability until servers get restored	Yes
	Available 24x7	Yes
Security	Role based authorization	Yes
Data Retention	Data Archival Period	TBD
Concurrency	How many concurrent users?	Maximum 25
Scalability	User Growth Per Year	10 per annum

Table 2 : Non Functional Requirement

3.2 Design

For the design stage of the project prototype method was chosen in order to give a clear understating to the Stakeholders about the project. This method was very helpful to get user feedback for the proposed solution at each module implementation.

3.2.1 Development Methodology

This system was built for the internal users of the factory, since it needs to cater for the user behavior and busy lives of the users. Where system need to be easy to understand for any internal user. Because of all these reasons conventional waterfall method was not used and potential users of the system were involved throughout the lifecycle of the project by using the prototype method.

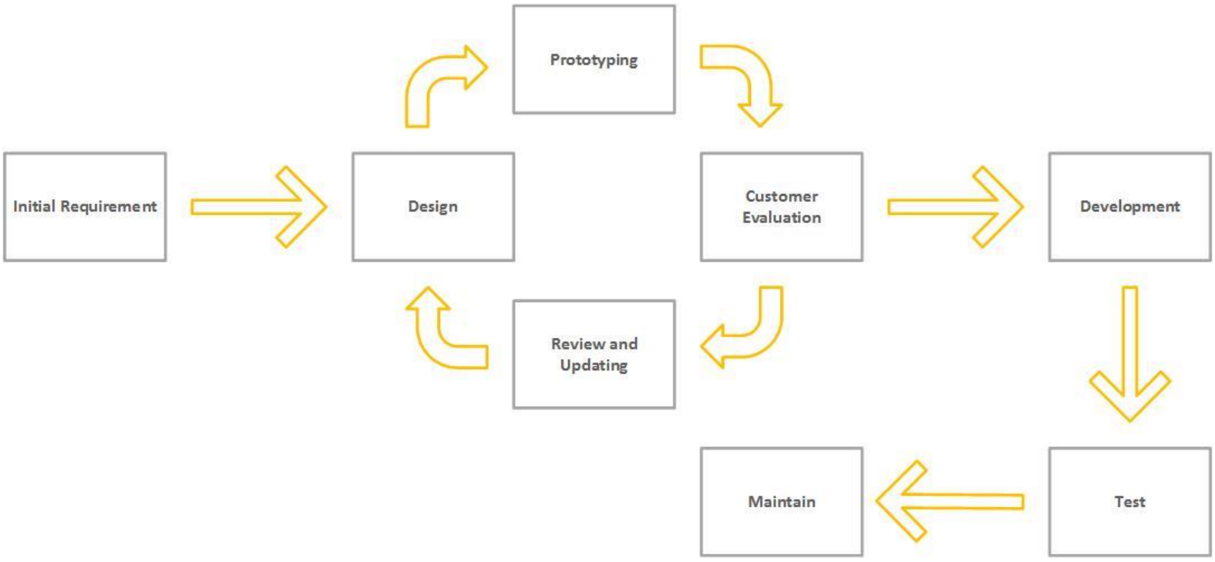


Figure 7 : Development Methodology

In the beginning to give the basic idea of the screens wireframes were used. Please refer Appendix A for the wireframes.

3.2.2 Use Case Diagram

Use case diagrams are used to analyze the system requirements. Use case provides the interaction between users and the system. Actors can be anyone or anything that interacts with the system. Also use case diagram presents the relationship between actors.

Proposed solution contain 3 Actors who are communicating with the system, following is the use case diagram for the proposed solution.

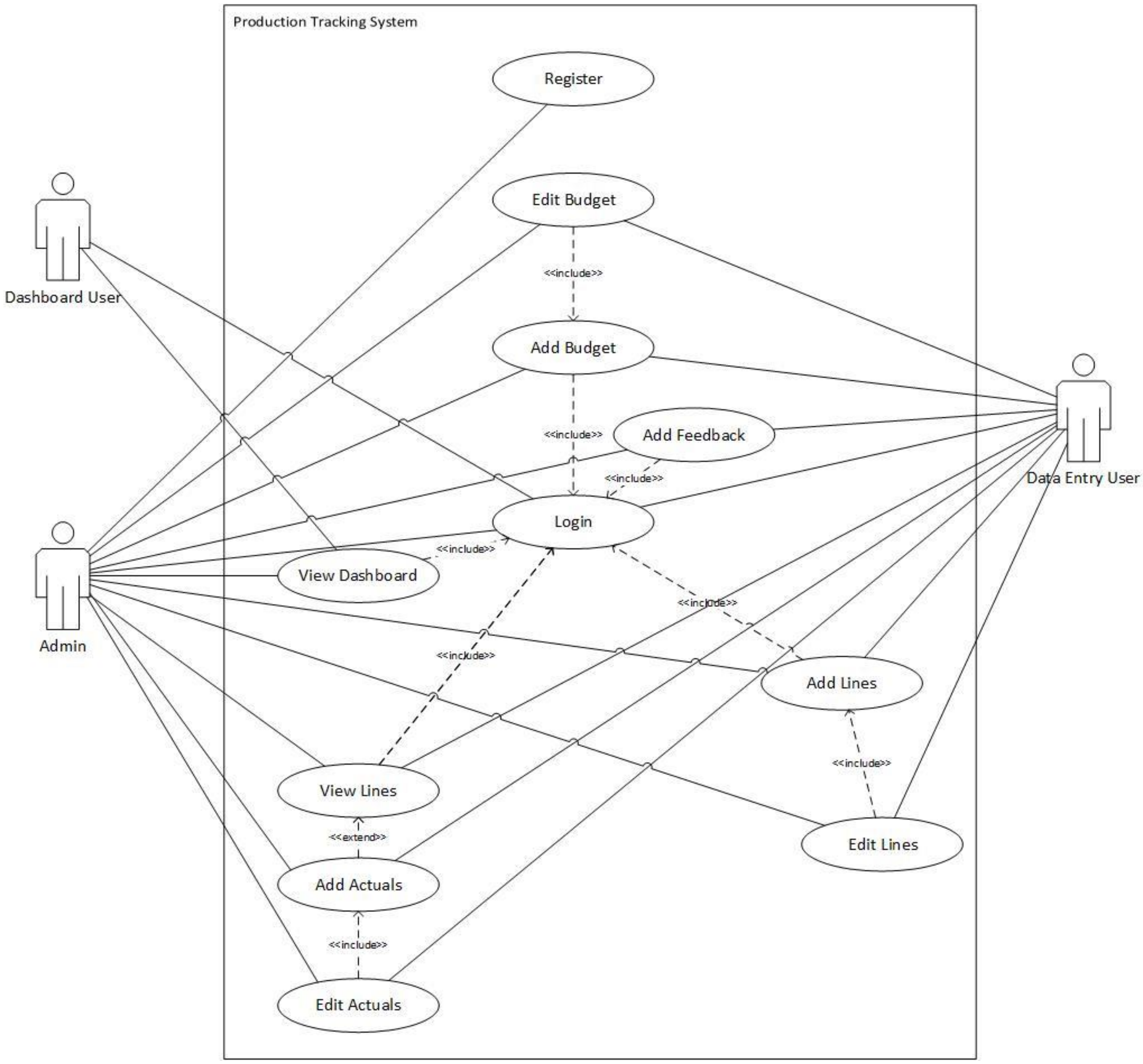


Figure 8 : Use case

Use case Name :	Register
Use case ID :	01
Pre-conditions :	Nil
Primary Actor :	Admin
Description :	Admin will create new user accounts using Register option
Trigger :	When user click the submit button
Typical flow of events :	• Admin enter new user e-mail account • Admin enter a password • Enter the submit button
Alternate courses :	Nil
Post-conditions :	Nil

Table 3 : Register use case

Use case Name :	Add Budget
Use case ID :	02
Pre-conditions :	User logged into the system
Primary Actor :	Data Entry User
Description :	Data Entry User should be able to add monthly budget data
Trigger :	When user click the submit button
Typical flow of events :	• User upload monthly planned budgeted details • Enter the submit button
Alternate courses :	Nil
Post-conditions :	Nil

Table 4 : Add budget use case

Use case Name :	Edit Budget
Use case ID :	03
Pre-conditions :	Data Entry User should be logged in to the system
Primary Actor :	Data Entry User
Description :	Data Entry User should be able to edit monthly planning budget data
Trigger :	When user clicks edit button
Typical flow of events :	• User Search for details • User click edit button • User edit monthly planning budget data • Enter the submit button
Alternate courses :	Nil
Post-conditions :	Nil

Table 5 : Edit Budget use case

Use case Name :	View Dashboard
Use case ID :	04
Pre-conditions :	User should be logged into the system
Primary Actor :	Dashboard User
Description :	Dashboard User should be able to view the dashboard
Trigger :	When user clicks different KPI’s of the dashboard
Typical flow of events :	• User go to dashboard URL and login • User clicks on a KPI
Alternate courses :	Nil
Post-conditions :	Nil

Table 6 : View Dashboard use case

Use case Name :	Add Actuals
Use case ID :	05
Pre-conditions :	Data Entry User should be logged in to the system
Primary Actor :	Data Entry User
Description :	Data Entry User should be able to Add hourly production details
Trigger :	When user clicks submit button
Typical flow of events :	• User go to hourly entry form • User enter hourly details for the lines • User clicks submit button
Alternate courses :	Nil
Post-conditions :	Nil

Table 7 : Add actuals use case

Use case Name :	Edit Actuals
Use case ID :	06
Pre-conditions :	Data Entry User should be able to Add hourly production details
Primary Actor :	Data Entry User
Description :	Data Entry User should be able to edit hourly production actual details
Trigger :	When user clicks submit button
Typical flow of events :	• User go to hourly entry form • User edit hourly details for the lines • User clicks submit button
Alternate courses :	Add hourly actual details
Post-conditions :	Nil

Table 8 : Edit Actuals use case

Use case Name :	Add Lines
Use case ID :	07
Pre-conditions :	Data Entry User should be logged in to the system
Primary Actor :	Data Entry User
Description :	Data Entry User should be able to add new lines to the system
Trigger :	When user clicks submit button
Typical flow of events :	• User go to add lines form • Add new lines • User clicks on submit button
Alternate courses :	Nil
Post-conditions :	Nil

Table 9 : Add Lines use case

Use case Name :	Edit Lines
Use case ID :	08
Pre-conditions :	Data Entry User should be logged in to the system
Primary Actor :	Data Entry User
Description :	System Admin should be able to edit lines to the system
Trigger :	When user clicks submit button
Typical flow of events :	• User go to add lines • Edit Lines • User clicks on submit button
Alternate courses :	Nil
Post-conditions :	Add Lines

Table 10 : Edit Lines use case

Use case Name :	View Lines
Use case ID :	09
Pre-conditions :	Data Entry User should be logged in to the system
Primary Actor :	Data Entry User
Description :	Data Entry User should be able to view lines
Trigger :	Nil
Typical flow of events :	• Data Entry User go to view lines form • View line details
Alternate courses :	Edit lines
Post-conditions :	Add lines

Table 11 : View lines use case

Use case Name :	Add Feedback
Use case ID :	10
Pre-conditions :	Data Entry User should be logged in to the system
Primary Actor :	Data Entry User
Description :	Data Entry User should be able to add feedback
Trigger :	When user clicks submit button
Typical flow of events :	• User go to feedback form • Add line feedback • User clicks submit button
Alternate courses :	Nil
Post-conditions :	Nil

Table 12 : Edit master data use case

Use case Name :	Login
Use case ID :	12
Pre-conditions :	User should be registered in the system
Primary Actor :	Admin, Data entry user & Dashboard User
Description :	Users should be able to log into the system
Trigger :	When user clicks submit button
Typical flow of events :	• User go to login form • Enter e-mail and password • User clicks submit button
Alternate courses :	Nil
Post-conditions :	Register

Table 13 : Login use case

3.2.3 User Characteristics

- User should be familiar with BCS production tracking process
- The user should have the basic computer literacy.

3.2.4 Principle Actors

- Admin
- Data Enter User
- Dashboard User

3.2.3 Activity Diagram

UML Activity Diagrams shows the flow of the processing in the each module of the system where it's mainly show how the flow of activates for each scenarios will work and how the decisions are made in each activity.

Figure 9 show the Activity diagram for User registration, in this diagram User and System are the two actors who are involved.

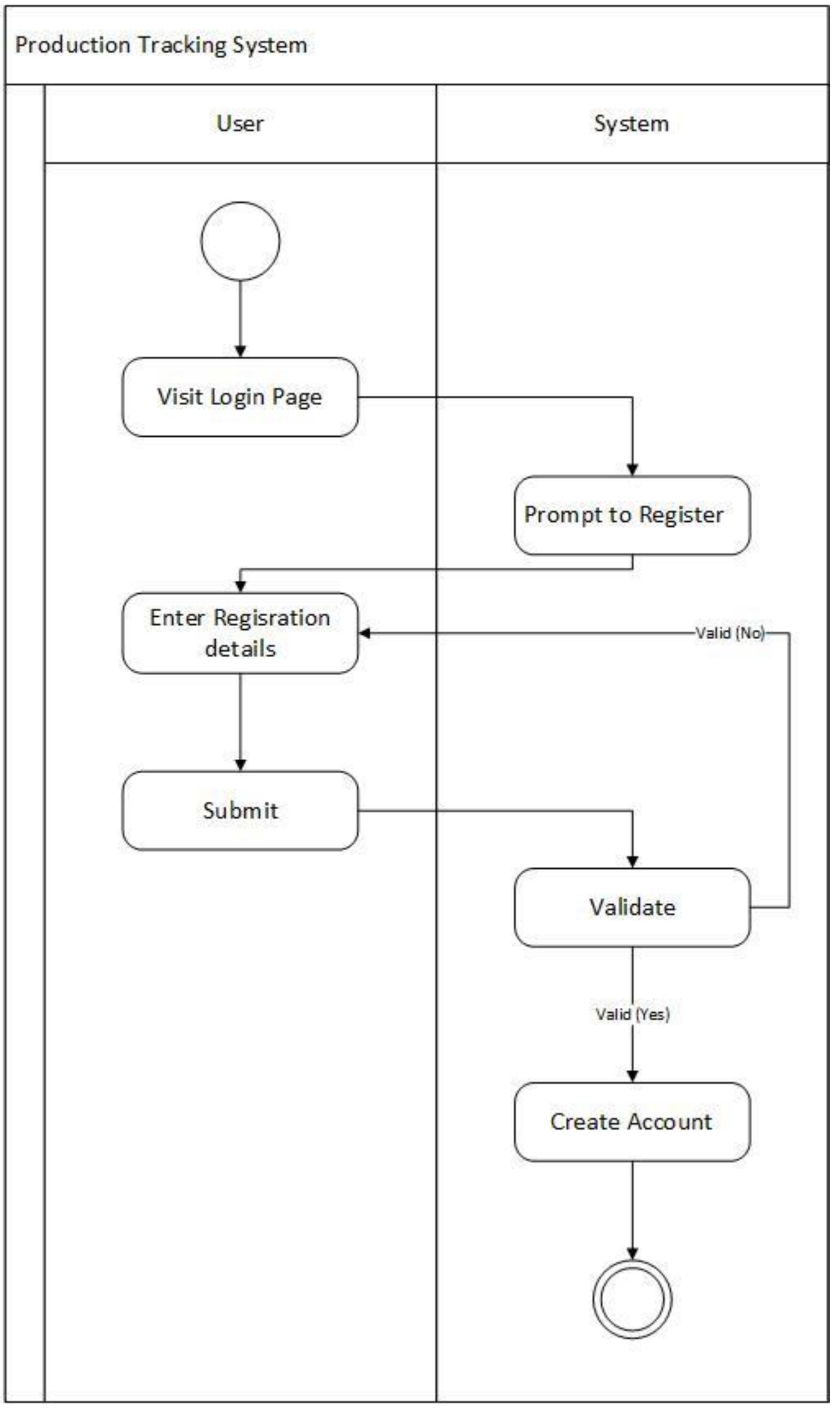


Figure 9 : User registration Activity diagram

3.2.5 Sequence Diagram

UML Sequence diagram is an interaction diagram which shows how processes operate with one another in which order and it shows how the messages are passed through the system.

Below are some sequence diagrams of the proposed solution.

Figure 10 shows the sequence diagram for the use case communication between Admin and View dashboard

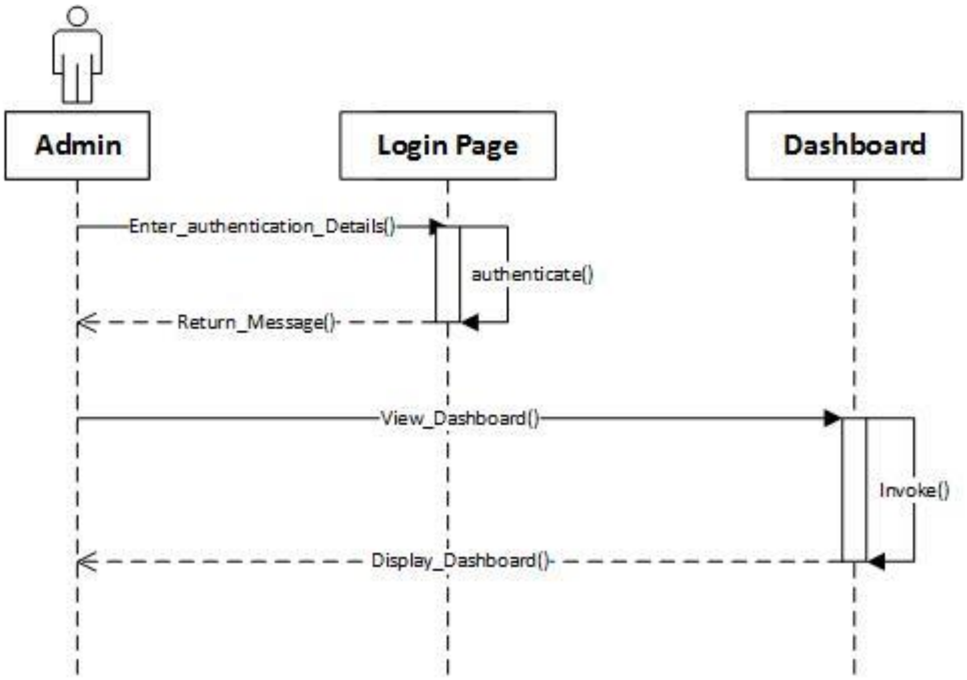


Figure 10 : Admin and Dashboard Sequence diagram

Figure 11 shows the sequence diagram for the use case communication between Admin and Add Lines

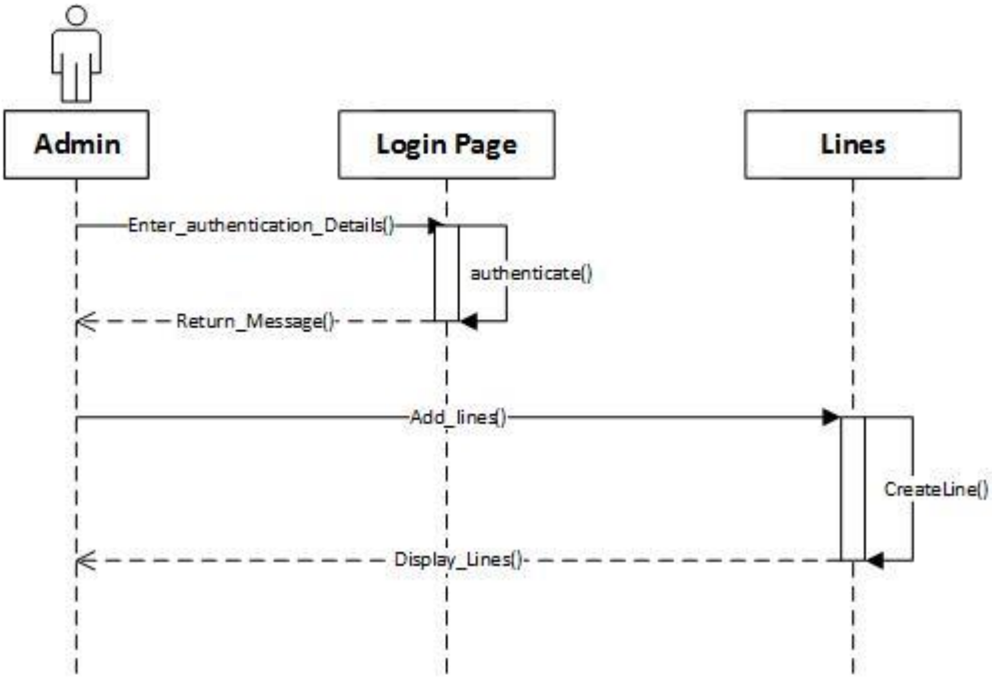


Figure 11 : Admin and add lines Sequence diagram

Figure 12 shows the sequence diagram for the use case communication between Data Entry User and Add Budget

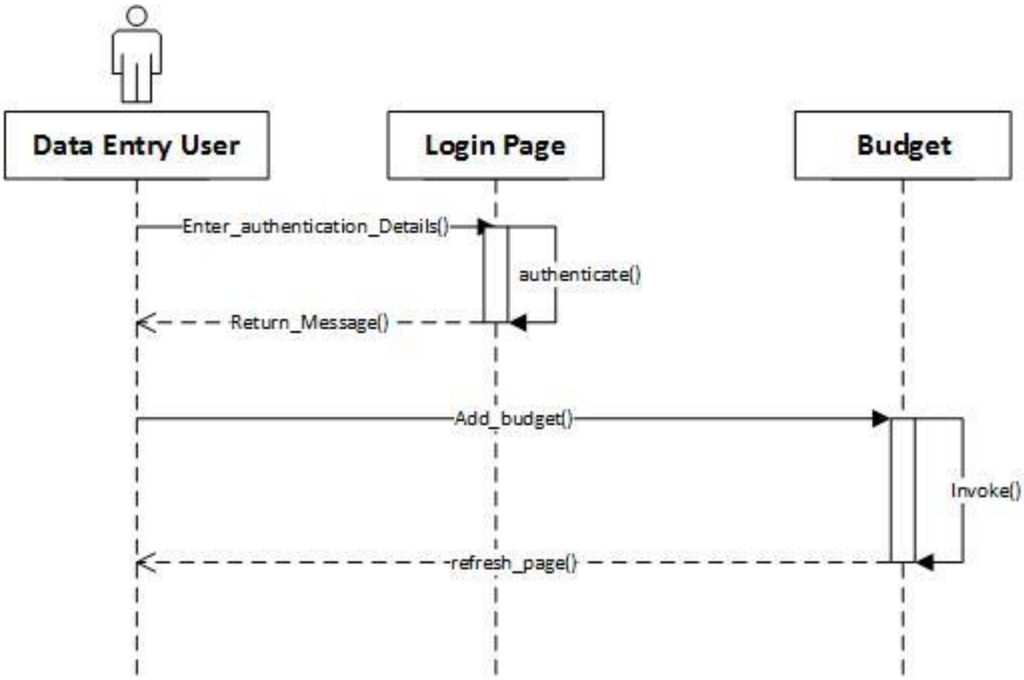


Figure 12 : Data entry user and Add budget Sequence diagram

3.2.4 ER Diagram

ER model defines the conceptual vies of a Data base based on the real-world entities and the associations of the system. ER model is considered as the first step or the best option for designing databases.

Figure 13 shows the Initial ER diagram for the proposed Production Tracking System.

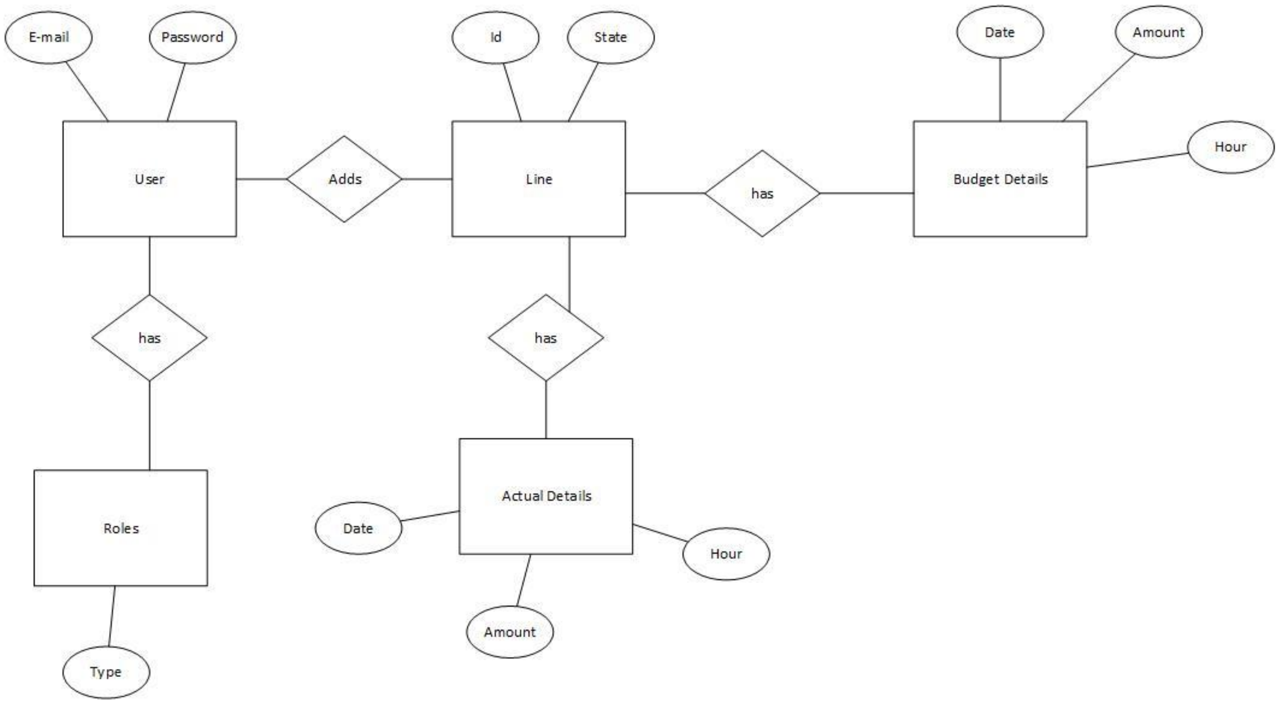


Figure 13 : ER diagram

Chapter 4: Implementation

4.1 Introduction

This chapter describes the technical implementation of the proposed Production tracking system. It will describe the about the technical background of the system and the development tools used for the development of the system. Future this chapter will elaborate the structure of the system and the architecture of the system as well.

4.2 Technical Background

This Production Tracking System is a web based solution. The complete workflow of production tracking and detail storing is done via this system. This is a secure system which requires users to create separate logins and manage and view relevant details.

4.3 Development Tools

Development of this system done using following programming languages, tools and techniques.

Following Programing languages were used for the development of this system:

- ASP.NET
- C#

Following Tools were used for the development of this system:

- Microsoft Visual Studio 2013
- Microsoft SQL Server 2008 R2
- Kendo UI

Kendo UI

The Kendo UI is a framework which contains HTML5, jQuery- based tools to build modern web applications. Kendo UI framework is developed by a company named Telerik. This framework ensures the responsive and it gives the standard charting options to the system.

There were few reasons to pick the above mentioned technology and tools,

- Customer preferred framework was Microsoft Dot net framework

- Customer was very concerned about the security of the system, compared to other open source framework platforms Microsoft platform is more secure.
- Responsive features in the Kendo UI charting control

4.4 Database Scripts

Given below are some data base scripts which was used to create the proposed solution database.

```
USE [MITProductionTracking]

GO

/****** Object:  Schema [app]    Script Date: 11/1/2015 9:12:33 PM ******/

CREATE SCHEMA [app]

GO

/****** Object:  Table [app].[Hours]    Script Date: 11/1/2015 9:12:33 PM ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [app].[Hours](

	[Id] [int] IDENTITY(1,1) NOT NULL,

	[Description] [varchar](50) NOT NULL,

	[From] [time](7) NOT NULL,

	[To] [time](7) NOT NULL,

	[StatusId] [int] NOT NULL,

 CONSTRAINT [PK_Hours] PRIMARY KEY CLUSTERED

(

	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]
```

Please Refer “Appendix B” for further database scripts

4.4 Module Structure of the System

The proposed solution was broken into two main modules,

Dashboard Module: Which contains all the KPI details were users can drill down and check line wise details

Data Entry Module: Which contains all the production details in the system, including Budget and Actual details of each line.

And to build the above mentioned modules Module-View-Controller software architecture pattern has been used. According to W3schools [4] MVC model defines web applications with 3 logic layers. Following diagram shows the 3 layers

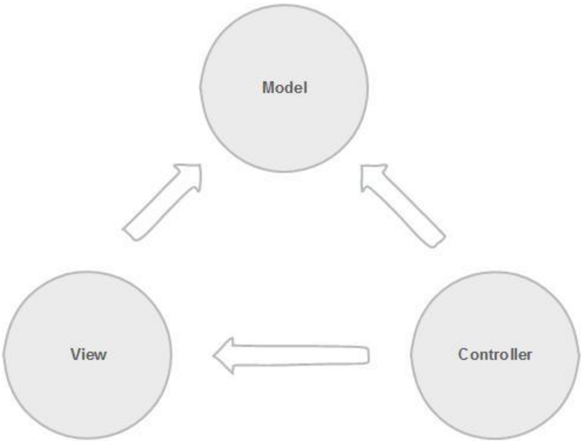


Figure 14 : MVC model diagram

4.4.1 Business Layer

This layer is normally known as the “Model layer” in the MVC architecture pattern. This layer handle the logic of the application.

4.4.2 Display Layer

This layer is normally known as the “View Logic” in the MVC architecture pattern. This layer handles the display of the data in the application.

4.4.3 Input Control

This layer is normally known as the “Controller logic” in the MVC architecture pattern. This layer handles the user interaction in the application.

Normally controllers read data from the view and control user input and send input data to the model

Chapter 5: Evaluation

Testing is a very important part of every development methodology to identify and fix the issues in that system. This chapter elaborates about the Testing aspect of the project, were the process followed and the levels of testing and the test cases used for the testing and also the User evaluation of the system is discussed in this capture.

5.1 Test Process

In the testing process two main techniques were used to test the system. Given below are the two main techniques and there advantages and disadvantages,

5.1.1 White Box Testing

According to many Softwaretestingfundamentals.com [5] White box testing refers to the testing conducted in the internal components or the structure of the program to check the proper execution of instructions and to check for any runtime errors which can occur during the execution of the program. Furthermore advantages of this type of testing include

1. Testing can be commenced during the early stages of development
2. Has the ability to cover most parts of the core system.

Disadvantages of White box testing include

1. Testing can get very complex and as a result of that high skilled resources are required with the need of people with in-depth knowledge to conduct the testing.
2. Managing the test scripts can become hectic if the number of changes done to the core during the testing becomes high.
3. With the fact that this method of testing is tightly linked with the testing aspect of the application, the tools that are required to do the development are required to be kept in stand-by-to- use all the time.

5.1.2 Black Box Testing

Black box testing take an external observation approach the applications functionality. The test designer creates the test cases where the user input and the expected output is checked with the actual regardless of internal functionality of the system.

As stated in Softwaretestingfundamentals.com [6] the base of the this method of testing lies in the selection of correct data inputs specific to the component and check it against the functional specifications for abnormal behavior on the output of data. As stated in Onestoptesting.com [7] following are the Advantages and Disadvantages of black box testing,

Advantages of black box testing include

- Tests are conducted from a users' point of view
- Test cases can be created as soon as the user specifications are created.
- Compared with the glass box testing it has a higher level of effect on large sections of code.

Disadvantages of black box testing include

- Only an average number of user inputs can be checked since checking for all possible inputs will be impossible time wise.
- Has a high possibility of missing out untested program paths

5.2 Testing Levels

During the implementation of the proposed solution different level of testing methods were used to ensure that each module is working without any issues. Following are the used levels,

5.2.1 Unit Testing

After each Module was completed unit testing was conducted to make sure that each unit has met the given requirements.

5.2.2 Integration Testing

When modules are integrated to each other integration testing was conducted to make sure all the modules are working after the integrations are done.

5.2.3 Validation Testing

Validation testing was conducted to make sure that all the validations are working properly.

5.2.4 System Testing

System testing was conducted to test the full functions of the system, this was conducted using black box testing method.

5.3 Test Cases

This section contains the Test case scenarios which were conducted for the Production Tracking System.

ID	Test Case	Steps	Expected Results	Actual Results	Status
1.0	**Login**				
	Login to the System	Login with an Admin User	Admin user should be able to see Dashboard and Data entry both functions	Same	Passed
		Login with an Dashboard User	Dashboard User should be able to see the dashboard functions	Same	Passed
		Enter and invalid username	Warning message should appear in the login screen	Same	Passed
		Enter and invalid password	Warning message should appear in the login screen	Same	Passed
		Enter invalid username and Password	Warning message should appear in the login screen	Same	Passed
		Valid username and password	Login to the system with appropriate login access	Same	Passed
2.0	**Register**				
	New user registration	User enter e-mail without a domain	Warning message shows up	Same	Passed

		User enter correct email & a password with letters only	Warning message shows up	Same	Passed
		User enter correct email & a password with numbers only	Warning message shows up	Same	Passed
		User enter correct email and a password with numeric and non-numeric combinations, without any upper case letters	Warning message shows up	Same	Passed
		User enter correct email and a password with numeric and non-numeric combinations, with upper case letters	User gets registered	Same	Passed
3.0	**View Dashboard**				
	View Dashboard KPIs	Login and Check on KPI 1	KPI goes to next level	Same	Passed
		Login and Check on KPI 2	KPI goes to next level	Same	Passed
		Login and Check on KPI 3	KPI goes to next level	Same	Passed
		Login and Check on KPI 4	KPI goes to next level	Same	Passed
4.0	**Add Budget**				
	Add Budget details	Upload the budget with correct upload format correct details	details upload to the system	Same	Passed

		Upload the budget with correct upload format wrong details	warning message should appear	Same	Passed
		Upload the budget with In-correct upload format correct details	warning message should appear	Same	Passed
		Upload the budget with In-correct upload format in-correct details	warning message should appear	Same	Passed

Table 14 : Test Cases

Please Refer “Appendix C” for further Test Cases

5.4 Test Data

Test data were selected were based on possible user inputs and selected past factory line details. The data that were collected can be divided into following categories,

1. Correct Data
2. Incorrect data
3. Partially correct data
4. Wrong data types
5. Incomplete data

When above categories considered with the login scenario following are few examples,

1. Correct Data – Input all the details correctly
2. Incorrect data – Typing wrong mail id and wrong password
3. Partially correct data – Typing correct mail id and wrong password
4. Wrong data types – typing mail id to the password field
5. Incomplete data – typing mail id without domain name (e.g: @gmail.com, @hotmail.com)

When above categories considered with the budget data entry scenario following are few examples,

1. Correct Data – Upload correct details using correct format
2. Incorrect data – Upload wrong details using wrong format
3. Partially correct data – Upload wrong details using correct format
4. Wrong data types – Enter Hour number to the line number in the uploading file format
5. Incomplete data – Enter details which out the date in the uploading file format

5.5 User Evaluation

For User evaluation User acceptance testing (UAT) method was used in this project. After each module was developed to check if the system is meeting the user requirements and expectations of the User the User acceptance testing (UAT) was performed. During the implementation after each incremental module was shared with the stakeholders as a prototype and defects and issues raised by then were fixed along with the next incremental version of the system. For the UAT high level Test cases of each module was used. Given below are the test cases which was used for UAT,

Scenario	Description	Expected Results	Actual Results	Status
Login				
Login to the System	Login with an Admin User	Admin user should be able to see Dashboard and Data entry both functions	Admin user see Dashboard and Data entry both functions	Pass
	Login with an Dashboard User	Dashboard User should be able to see the dashboard functions	Dashboard User see the dashboard functions	Pass
	Valid username and password	Login to the system with appropriate login access	User able to login to the system	Pass
Register				
New user registration	User enter e-mail without a domain	Warning message shows up	Warning message shows up	Pass
	User enter correct email and a password with numeric and non-numeric combinations, with upper case letters	User gets registered	User gets registered	Pass
View Dashboard				

View Dashboard KPIs	Login and Check on KPI 1	KPI goes to next level	KPI goes to next level	Pass
Add Budget				
Add Budget details	Upload the budget with correct upload format correct details	details upload to the system	details upload to the system	Pass
	Upload the budget with In-correct upload format in-correct details	warning message should appear	warning message should appear	Pass
Add Actuals				
Add Actual details	Upload the Actual with correct upload format correct details	details upload to the system	details upload to the system	Pass
	Upload the Actual with In-correct upload format in-correct details	warning message should appear	warning message should appear	Pass
Add Lines				
Add new lines	Add a line with numeric value as the line name	line gets added	line gets added	Pass
	Add a line with non-numeric value as the line name	line gets added	line gets added	Pass
Edit Budget				
Edit production budget	Click the edit button in the table and change the value and hit save button	New value get added	New value get added	Pass
	Click the edit button in the table and change the value to a non-numeric value and hit the save button	Warning message shows up	Warning message shows up	Pass
Edit Actuals				

Edit production Actuals	Click the edit button in the table and change the value and hit save button	New value get added	New value get added	Pass
	Click the edit button in the table and change the value to a non-numeric value and hit the save button	Warning message shows up	Warning message shows up	Pass
Edit Lines				
Edit lines	Click the edit button change the status ID to 2	Line status changed to disable	Line status changed to disable	Pass
	Click the edit button change the status ID to 1	Line status changed to enable	Line status changed to enable	Pass
Add Feedback				
Add line feedback	Select a line and add feedback and save	Details get saved to the system	Details get saved to the system	Pass
	Click save button without adding details	Warning message shows up	Warning message shows up	Pass
View Lines				
View the active lines	Click the edit button	Go to edit form	Go to edit form	Pass

Table 15 : UAT Test case

Chapter 6: Conclusion and Future Work

6.1 Introduction

This chapter contains the conclusion of the project were Results with respect to the objectives were final outcome and objectives will be compared, Critical appraisal were how overall project has helped the business of the factory and finally future work to improve the project more will be discussed in this chapter.

6.2 Results with Respect to the Objectives

Objectives of the project was discussed in previous chapters. Main objective of the project was to provide a line Production detail storing method and a way of displaying those details to the management. All the scope mentioned objectives have been achieved in the project.

6.3 Critical Appraisal

Any organization manual work is hard to track and it will be always it will be messy. With the help of production tracking system the manual work of Production tracking in Brandix Casualwear Seeduwa was eliminated. This system will provide more business critical information to management to make managerial decisions. Also with this system now all the production details can be viewed immediately without going through many excel files. Also budgeted and Actual details can be compared hourly easily. And line feedback will reach to management quickly were previously it took days for it to reach the management.

6.4 Future Work

Production tracking system of Brandix Casualwear Seeduwa was developed based on the customer requirements. Following points have been identified as the future work of this project.

- Change login user id in site master to display user’s first name instead of e-mail.
- Purchase Kendo UI license (Customer will buy)
- Create a mobile version of the solution
- E-mail notification system to inform registration details to new users.

Overall the project has met the objectives defined in the beginning of the project, although few new additions can be added to make the system more effective for the users and make their work more productive.

Reference

[1] Brandix.com, "Brandix Lanka Limited - End-to-end apparel solutions in Sri Lanka", 2015. [Online]. Available: http://brandix.com/. [Accessed: 28- Dec- 2015].

[2] Tableau Software, "Business Dashboards", 2015. [Online]. Available: http://www.tableau.com/solutions/business-dashboards. [Accessed: 28- Dec- 2015].

[3]G. TECHNOLOGIES, "RFID Shop floor Tracking, Feedback and Lean Production | GPRO Technologies", Gprotechnologies.com, 2015. [Online]. Available: http://www.gprotechnologies.com/products/gpro-sdt/. [Accessed: 28- Dec- 2015].

[4] W3schools.com, "ASP.NET MVC Introduction", 2015. [Online]. Available: http://www.w3schools.com/aspnet/mvc_intro.asp. [Accessed: 28- Dec- 2015].

[5] Softwaretestingfundamentals.com, "White Box Testing | Software Testing Fundamentals", 2010. [Online]. Available: http://softwaretestingfundamentals.com/white-box-testing/. [Accessed: 28- Dec- 2015].

[6] Softwaretestingfundamentals.com, "Black Box Testing | Software Testing Fundamentals", 2010. [Online]. Available: http://softwaretestingfundamentals.com/black-box-testing/. [Accessed: 28- Dec- 2015].

[7] Onestoptesting.com, "Black Box Testing", 2015. [Online]. Available: http://www.onestoptesting.com/blackbox-testing/. [Accessed: 28- Dec- 2015].

[8] Onestoptesting.com, "Advantages and Disadvantages", 2015. [Online]. Available: http://www.onestoptesting.com/blackbox-testing/advantages-disadvantages.asp. [Accessed: 28- Dec- 2015].

Appendices

Appendix A: Wire frames

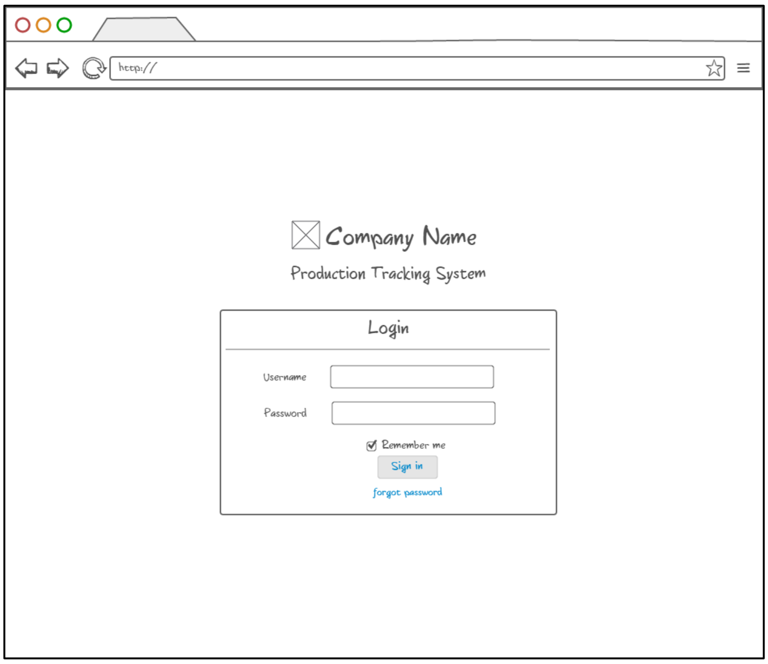
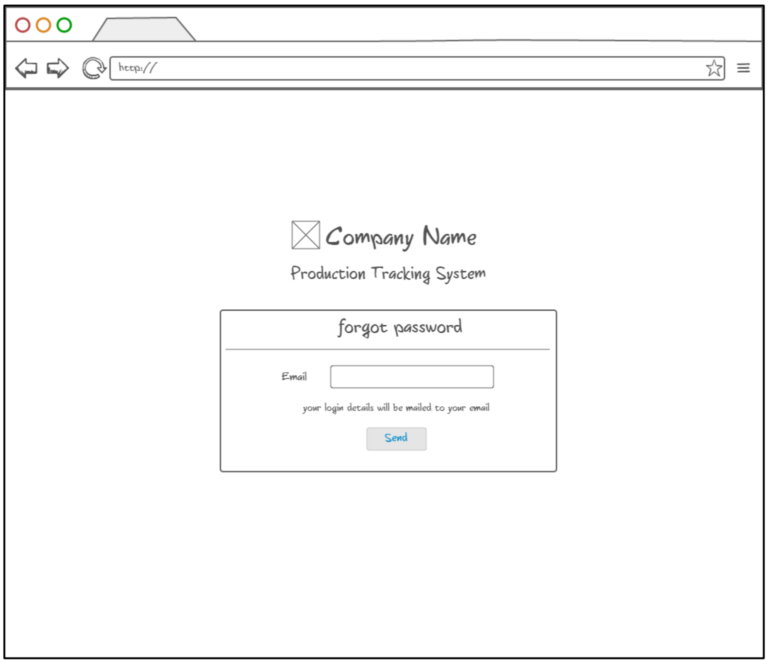


Figure 15



your login details will be mailed to your email

Figure 16

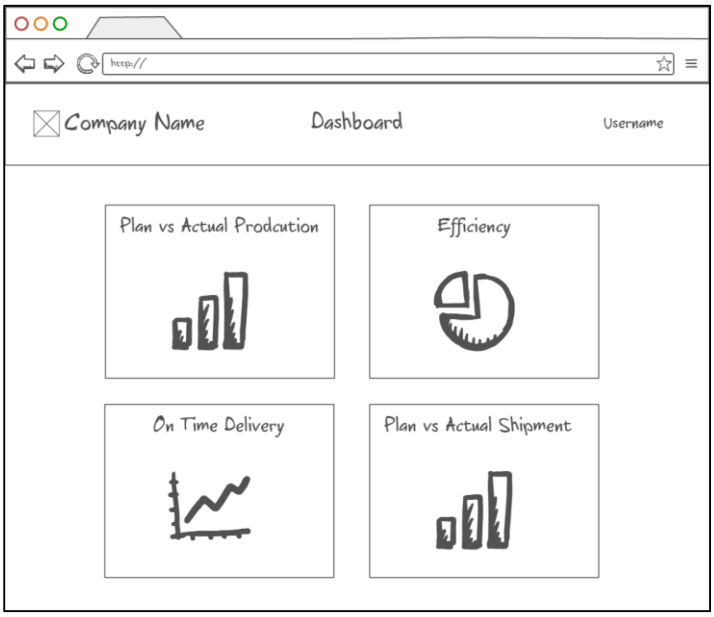


Figure 17

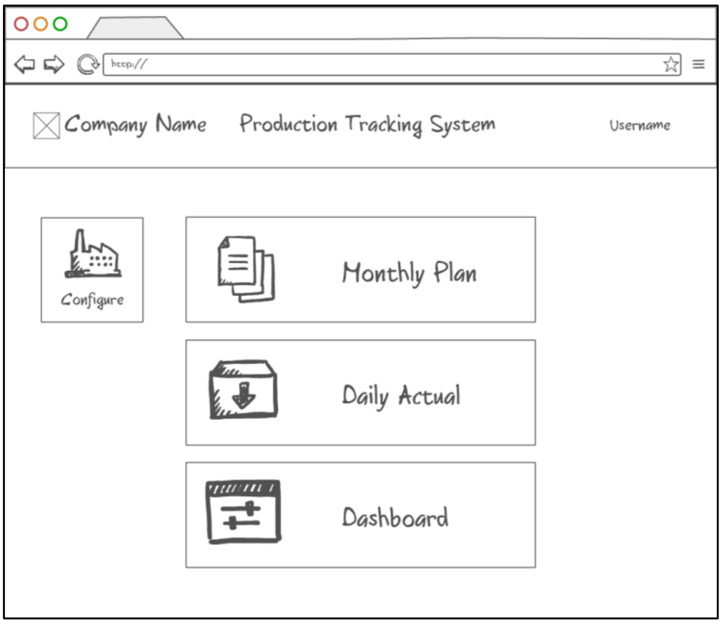


Figure 18

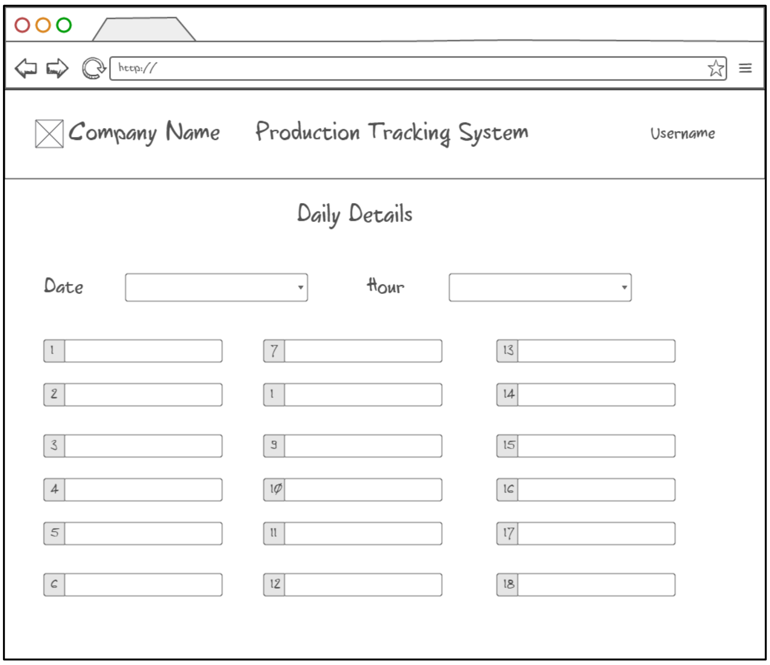


Figure 19

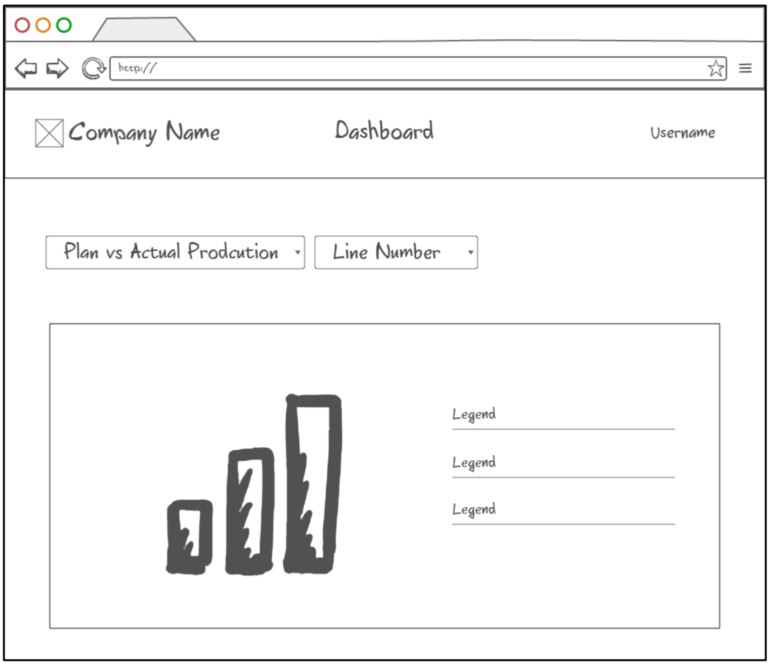


Figure 20

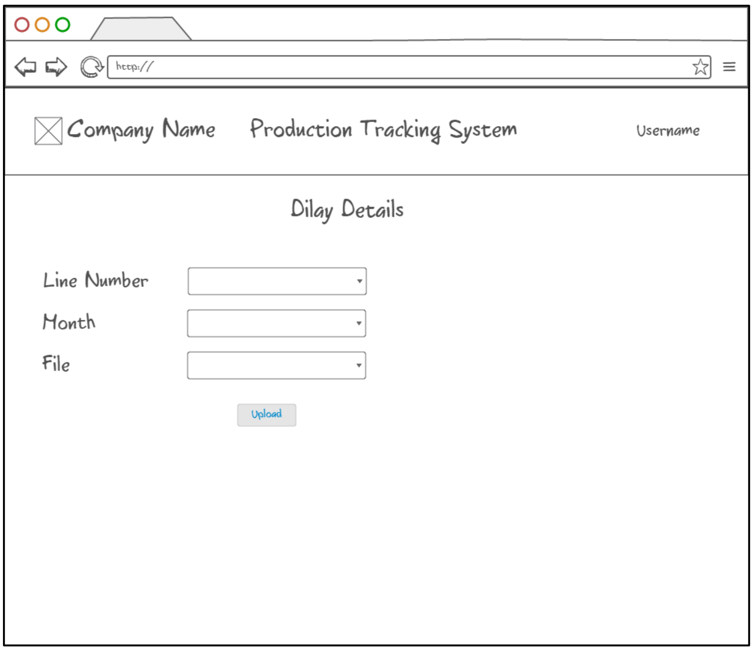
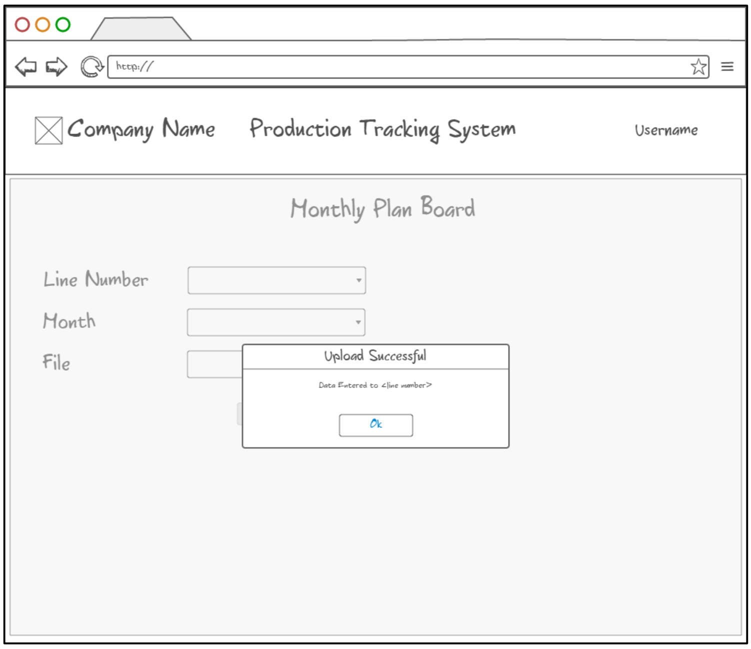


Figure 21



Upload Successful

Data Entered to <line number>

Figure 22

Appendix B: Database Scripts

```
USE [MITProductionTracking_Sec]

GO

/****** Object:  Schema [app]    Script Date: 11/1/2015 9:14:19 PM ******/

CREATE SCHEMA [app]

GO

/****** Object:  Table [app].[Hours]    Script Date: 11/1/2015 9:14:19 PM ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [app].[Hours](

	[Id] [int] IDENTITY(1,1) NOT NULL,

	[Description] [nvarchar](50) NOT NULL,

	[From] [time](7) NOT NULL,

	[To] [time](7) NOT NULL,

	[StatusId] [int] NOT NULL,

 CONSTRAINT [PK_app.Hours] PRIMARY KEY CLUSTERED

(

	[Id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]

) ON [PRIMARY]
```

```
GO

/****** Object:  Table [app].[ProductionLines]    Script Date: 11/1/2015 9:14:19 PM ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [app].[ProductionLines](

	[Id] [int] IDENTITY(1,1) NOT NULL,

	[Description] [nvarchar](50) NOT NULL,

	[StatusId] [int] NOT NULL,

 CONSTRAINT [PK_app.ProductionLines] PRIMARY KEY CLUSTERED

(

	[Id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]

) ON [PRIMARY]



GO

/****** Object:  Table [app].[Status]    Script Date: 11/1/2015 9:14:19 PM ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO
```

```
CREATE TABLE [app].[Status](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Description] [nvarchar](50) NOT NULL,
 CONSTRAINT [PK_app.Status] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]


GO
/****** Object:  Table [dbo].[__MigrationHistory]    Script Date: 11/1/2015 9:14:19 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[__MigrationHistory](
	[MigrationId] [nvarchar](150) NOT NULL,
	[ContextKey] [nvarchar](300) NOT NULL,
	[Model] [varbinary](max) NOT NULL,
	[ProductVersion] [nvarchar](32) NOT NULL,
 CONSTRAINT [PK_dbo.__MigrationHistory] PRIMARY KEY CLUSTERED
(
	[MigrationId] ASC,
	[ContextKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[AspNetRoles]    Script Date: 11/1/2015 9:14:19 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetRoles](
	[Id] [nvarchar](128) NOT NULL,
	[Name] [nvarchar](256) NOT NULL,
 CONSTRAINT [PK_dbo.AspNetRoles] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]


GO

/****** Object:  Table [dbo].[AspNetUserClaims]    Script Date: 11/1/2015 9:14:19 PM ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AspNetUserClaims](

	[Id] [int] IDENTITY(1,1) NOT NULL,

	[UserId] [nvarchar](128) NOT NULL,

	[ClaimType] [nvarchar](max) NULL,

	[ClaimValue] [nvarchar](max) NULL,

 CONSTRAINT [PK_dbo.AspNetUserClaims] PRIMARY KEY CLUSTERED

(

	[Id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

```
GO
/****** Object:  Table [dbo].[AspNetUserLogins]    Script Date: 11/1/2015 9:14:19 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUserLogins](
	[LoginProvider] [nvarchar](128) NOT NULL,
	[ProviderKey] [nvarchar](128) NOT NULL,
	[UserId] [nvarchar](128) NOT NULL,
 CONSTRAINT [PK_dbo.AspNetUserLogins] PRIMARY KEY CLUSTERED 
(
	[LoginProvider] ASC,
	[ProviderKey] ASC,
	[UserId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/****** Object:  Table [dbo].[AspNetUserRoles]    Script Date: 11/1/2015 9:14:19 PM ******/
SET ANSI_NULLS ON
```

```
GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AspNetUserRoles](

	[UserId] [nvarchar](128) NOT NULL,

	[RoleId] [nvarchar](128) NOT NULL,

 CONSTRAINT [PK_dbo.AspNetUserRoles] PRIMARY KEY CLUSTERED

(

	[UserId] ASC,

	[RoleId] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]


GO

/****** Object:  Table [dbo].[AspNetUsers]    Script Date: 11/1/2015 9:14:19 PM ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AspNetUsers](

	[Id] [nvarchar](128) NOT NULL,

	[Email] [nvarchar](256) NULL,
	[EmailConfirmed] [bit] NOT NULL,

	[PasswordHash] [nvarchar](max) NULL,

	[SecurityStamp] [nvarchar](max) NULL,

	[PhoneNumber] [nvarchar](max) NULL,

	[PhoneNumberConfirmed] [bit] NOT NULL,

	[TwoFactorEnabled] [bit] NOT NULL,

	[LockoutEndDateUtc] [datetime] NULL,

	[LockoutEnabled] [bit] NOT NULL,

	[AccessFailedCount] [int] NOT NULL,

	[UserName] [nvarchar](256) NOT NULL,

 CONSTRAINT [PK_dbo.AspNetUsers] PRIMARY KEY CLUSTERED

(

	[Id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]


GO

INSERT [dbo].[__MigrationHistory] ([MigrationId], [ContextKey], [Model],
[ProductVersion]) VALUES (N'201511011243176_InitialCreate',
N'MITProductionTracking.Models.ApplicationDbContext',
0x1F8B0800000000000400ED5D5B6FE3B8157E2FD0FF20E8A92DB2762E9DE934B0779
175926ED0DC30F62CFA16D012E30823515A89CA2628FACBFAD09FD4BF50521225F1
A69B6559DE0E1658C422F99DC3C373C8C3431ECE7FFFFD9FD90F6F9E6BBCC230727
C34374F26C7A60191E5DB0EDACCCD183F7FF7C9FCE1FBDFFF6E76657B6FC6CFAC
```

```
DE19AD475AA2686EBE601C9C4FA791F5023D104D3CC70AFDC87FC613CBF7A6C0F6
A7A7C7C77F9D9E9C4C218130099661CC3EC7083B1E4C7E909F0B1F5930C03170EF7C
1BBA51F69D942C1354E31E78300A8005E7E6DDCDEA31F4EDD8C284935508ACAF84
D549DACE342E5C07109E96D07D360D80908F01AD77FE25824B1CFA68B30CC807E0A
EDE0348EA3D033782594FCE8BEA4D3B757C4A3B352D1A32282B8EB0EFB5043C39C
BA434159B7792B5994B91C8F18AC81BBFD35E27B29C9B850C6F1D444420923C5FB82
1AD5E2DF0895C221550FC23438972946B14513CFADF91B1885D1C87708E608C43E01E
198FF1DA75ACBFC3F795FF15A2398A5DB7DC33D23752C67D209F08A90086F8FD337
CCEFA7B639BC6946F37151BE6CD4A6D5219DC207C766A1AF7843858BB30579C92BC
96D80FE1DF208221C0D07E0418C310510C98885EA22ED0BA8491153A413ADA2951A2
B04446A67107DE6E21DAE097B9F98198DDB5F3066DF621E3E30B7288CD9236388CA1
82CF6ADA4BD28538AAEFAD80720F5E9D4DD279259E697C866E521CBD38416A9393
B4E889D70E52F53AF4BDCFBE9B37166B3CAD40B8819870E857565BFA716809ACCE
A685F657DA04E3BB6F5B4871BFD9C0A86C40ABBD3F1115522B2F2D79624A52686CE
9B3A4A6E532956E56312219499FF6C498A9B127D69F4EF6443BDFBF3551D46FB6342A
5B126853E5634457C4CD6B0DB0F2B76A3EE07AD6604A102D4D355D74322FA60A14B
6D2CC722FF1220AEE219EB0869314F23A2470BFFAE1D74919919858D37685D99D3635
BBB393F5F3D9A70F1F817DF6F1CFF0ECC3F026A8B08693D34FBB3007FAFF0AAAA7
1F3EEE7641237B1FCD82561EEFA7AC5AA1C472A9A4CB8A2ABDA83485EA5FAD19E
AF8559B722AABB7B22AED50174B602486B606C6EF6EE936D6B88B20208397A816954
8677745C039740765387DB8F280E3F6303D36A0B2F0D1B3137A30EFE58F3E5146805AF
3FC08A288CC0EF64F207AA9609DFCD903EB4B68C521515AE23278C1CEA93DBEF808
DEC7DE9ADAC270B47A1B9AD5AFFE35B088EF7C8568ABADF16E7DEBAB1FE32B6
45F123FFC0BB61820FD997AA70D017A61E7C2B260145D136586F6C28F116EE9E32A1
6817DBB270B17389EDA3F1126D52756B5F051D435243F45534DE5AB54B17AEB6F1C
D48C555655CF6A5AA396D5AC5A5B562958334EB39A7A46930AB57CA6B57AF3FE92
11EADFFD4B60C7EFFF8D3DBAB02FE731193E4A74E76B5342E967E0C67D93EA640D
C924D0BF3524B0E3B786844DF2F9D5B1A957D26053C42A13F846F5D5FBAD7A9B133
81BDA1CB86E0E4D7C983940672E1751E45B4E6205A568330BC5F18C13E7CDA888CB
A5FCB3681EE9025169872E6E84ECDCFC932407355C1EF92FE0585C90073C31457D7C4
097D085181A17567A58BC0091056C7904882C6CFE0B5161185263057477131113741096
F5DD4196130057CFB6D0A4E1A24459CAC1C5924B1840442712BDEC9B502D82B932E
D9C8420A63AA9CCA625F5A9D62ACD018B4E23EA4E5B06558E5AB6143A2BDE0BA8
33864E0A582DA50174B15A1E07A0968AC8AD6EECABC2B8C5B8F3870983286545F05
8C1581112DE894AEAA534803AEA25D184B83A023C90226AB6E8BA31AFDBAF17E32
```

```
E056607D1C99A4081462FB3CDEA4E14B35A62032867B5489A30A03DCDD8878266819
9A60A204669C6A6A0427848A3A0D9FE711005E525B60705E54572700A9AC6E39A8EB
F109C1B9B7AF251C1E197F54A71ED413739798C4C35D38D366983490B18CAEA79B9A
685F00D2B225184CF2C181565FB7A514528F81262DDE5B6629BAFD98948BB701E90E
DA7241C5650D33EBBF627354F6302358DB318BAD498F7AA6B404493A8022CCCA606
34BBD5210149D3430BE6D8314C2577994FD402961D9954C2662B99005BD2687E44F3B
B59A51A8AAB5BA27155478CF24EE45A235967758CA804A0D1CE29DFA306BDD5DD
04953BDE24A4D126A851DB9B36A1881298D495AD65A4BAEF24CBA76E6FDD74775D
EA4A66CE1562A9D80C97701453C4D642D11DB2CA8269B2D76BB3DB2B752C1B8C0
A01D5ECCC3442629DE95D4A6CB2AA97926AC3D166CBB1959484ED81464AAC33BD
4B29D3D17A21299CDE166EEF5622E25DD49E8C8D1D5BE4DE545E369BA6396DD987
D95493FC36BB0341E0A04D29192EFB622CD34CB8C577CBF689615E8A31B538698BB
E5F4E09FB21D840A19490269C5E3B61842F01066B400F6D16B6275553FA8E1A8780915
4BB87F270321F81B5A37FA76D1BA615245EA5EC826778D7A4DF1EF5E3939372EDAA
240318346911B820541CCF2F7C37F6907E63A16FCD5DE72FC37005CDF18AF075194C1
FD4A68A2A4846DA904863226D1BF9A16EA4085A07ADE3F8ABF11A8CBBAEE1B8C6
7B4FA394BAC1BD0D52B2E76A3F44EA66E31A201D5E9A3353064ABF3447A04933E5F
6F4F76F7042D0790A1D558DDBA1B757B9EAE6BB51BDF4C266B97DFA653443243B4
EBD0F571EFFE83E647A089DE0592CAE2C7A5D7C4E8FC28EEBCA28BA23BCBD0DA
16EDBDB71D8C46853FB51AB45D88DAD65D9096580EC534B8CD2057709AC54D61C9
5CF412863F2252D96003ED1805B07F8A2165C96D3093826CB059DF0341255D768B18C
4A0904DCA22A95364756A41294A115C51DB0153C8B65CD5115D9066560457173EC22
F5409C4E47BC9A69235E3D2C6769C07CBBF54C83B19BB9B19FE5B074A5BB0C54FA
DC122BBBB42D8165DF47A957DA18610F7A959E986CA7571A0CFD6CC45D8BE627A3
CABBDC7A4CEEAE3337E157DDF5D6E3B5D3DE9DEA88142C14ABE4D4F3A0A1101
C9C6581BAFAE7B3A4C85D5AC5349818C962FF1E61E84D6885C9F21777E13A904EED
ACC21D40CE338C707ABFDF3C3D3E3915DEDD1ACF1B58D328B25D45A053FF10163
F6A0324EB3854B6B5E9382DEFE62B5EFF40AF20B45E4028BDFFB1E5E31A09FFD275
801B64C3B7B9F9CFA4D1B971F38F27D6EEC8780889169D1BC7C6BFFA4AE256DEFA
FFBF1FBB76EFF47C135E8B576D7092F8FB1842CB491F44FCCB166FDCF4003646A3D4
87C686D02EE5A82719495BBCDE5280D6C996769AFE957C3E326EA22FC8F9252605AB
308654C6628675BF3257C7B646FAF64873A9128D4D9BF2FABAE508F32F92B4E2266D
BA05379DDF29395C83E21EFC50A20A06D1FD7D8FB5837B79DB8371F9070FBCFDB1
2D6BCAF73BB64254BCD1D1175E2F22D4BDC1D1054BFBFE864D7E62E175B8269D55
BFC7D18535ED5B1CC9EABBE54B1CCDA721D6728F4B8D22EC74B01EE4B8D626E9C
```

```
183AD0C5D7ED4A005DC160F1774D08C03CBF9EF6D7554A4F4F786BD4FD5EE378F9
FDD33DE573E759A14D0FAE5802D92A587CAD03F84E4E74E39F9FB53163191653FD9
F6432850EDE5CAD1A952C73CFAB1A4CE176940FBCD981F3249BEE2EACF6F2A377E
04D99C8A7CAEFD67C00FAD6BBA73F991A711B7CB731F99B265597EFBCF661F5AD9
7487F52357B65639EB23D3B57DAD9F7BD6B4C64BE8DE33D0E564237158F99C62D1A
92E369E1AAF3BBD0231374140E3B3E916B62EF954245A9B7EAE22D222355D9B99AE
C25566256B73AF531D97E0F96246C65EFB3999F4B548756A9C8E5861545A8245153D51
7D4E9E48589A6024BA528D6AB2EDFA9A3946959DCDEA5493D564B256D1CED6C94
ADA599D6ADA9AFCD0DD65DDF3D6D434B5BC6C274542D001A4D6B7EDA87A8653
A5278E3E675E9971AB7AD9A266FDD65C5C54BE9131E61C79AE27358F74D4EDD52A
2F091F524A7C2F42E16643CD0DD7C3C980EF45247D9A4E8B8C77F9B22AF1194BFF22
2CF15B23675340D07F1F16418BF316F33A37E8D9674EABC011AB221C83DC410C6CE2
4A5E84D879061626C5F4203779BE38391CA3D709D6D0BE410F310E624CBA0CBDB5C
B9D2A51E7B78A7E92D6CFF33C7B48AEA5457D7481B0E9D003F007F463ECB876CEF7
B5E2E0450341BDEAECD8948E25A6C7A79BF71CE9DE17570F1D5026BE7C33B0825EE
012B0E8012DC12BECC21B51BF5BB801D67B71CCA603A91F085EECB34B076C42E045
1946D19EFC243A6C7B6FDFFF0F8F1721CC18790000, N'6.1.1-30610')

GO

INSERT [dbo].[AspNetUsers] ([Id], [Email], [EmailConfirmed], [PasswordHash],
[SecurityStamp], [PhoneNumber], [PhoneNumberConfirmed], [TwoFactorEnabled],
[LockoutEndDateUtc], [LockoutEnabled], [AccessFailedCount], [UserName]) VALUES
(N'bdb2b62f-7736-4ea8-838a-068537418a85', N'dln0925@gmail.com', 0,
N'AGKzEjoiF+7uhCZH5IC0SivGfF0j5PCRDtoZI29HD4IPDhjJ4pG+93yBi5mpHam8Ng=='
, N'5cf58780-f051-40d9-a5dd-d8dccb107273', NULL, 0, 0, NULL, 1, 0,
N'dln0925@gmail.com')

GO

ALTER TABLE [app].[Hours]  WITH CHECK ADD  CONSTRAINT
[FK_app.Hours_app.Status_StatusId] FOREIGN KEY([StatusId])

REFERENCES [app].[Status] ([Id])

ON DELETE CASCADE

GO
```

```
ALTER TABLE [app].[Hours] CHECK CONSTRAINT
[FK_app.Hours_app.Status_StatusId]

GO

ALTER TABLE [app].[ProductionLines]  WITH CHECK ADD  CONSTRAINT
[FK_app.ProductionLines_app.Status_StatusId] FOREIGN KEY([StatusId])

REFERENCES [app].[Status] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [app].[ProductionLines] CHECK CONSTRAINT
[FK_app.ProductionLines_app.Status_StatusId]

GO

ALTER TABLE [dbo].[AspNetUserClaims]  WITH CHECK ADD  CONSTRAINT
[FK_dbo.AspNetUserClaims_dbo.AspNetUsers_UserId] FOREIGN KEY([UserId])

REFERENCES [dbo].[AspNetUsers] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserClaims] CHECK CONSTRAINT
[FK_dbo.AspNetUserClaims_dbo.AspNetUsers_UserId]

GO

ALTER TABLE [dbo].[AspNetUserLogins]  WITH CHECK ADD  CONSTRAINT
[FK_dbo.AspNetUserLogins_dbo.AspNetUsers_UserId] FOREIGN KEY([UserId])

REFERENCES [dbo].[AspNetUsers] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserLogins] CHECK CONSTRAINT
[FK_dbo.AspNetUserLogins_dbo.AspNetUsers_UserId]
```

```
GO
ALTER TABLE [dbo].[AspNetUserRoles]  WITH CHECK ADD  CONSTRAINT
[FK_dbo.AspNetUserRoles_dbo.AspNetRoles_RoleId] FOREIGN KEY([RoleId])
REFERENCES [dbo].[AspNetRoles] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[AspNetUserRoles] CHECK CONSTRAINT
[FK_dbo.AspNetUserRoles_dbo.AspNetRoles_RoleId]
GO
ALTER TABLE [dbo].[AspNetUserRoles]  WITH CHECK ADD  CONSTRAINT
[FK_dbo.AspNetUserRoles_dbo.AspNetUsers_UserId] FOREIGN KEY([UserId])
REFERENCES [dbo].[AspNetUsers] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[AspNetUserRoles] CHECK CONSTRAINT
[FK_dbo.AspNetUserRoles_dbo.AspNetUsers_UserId]
GO
```

```sql
USE [MITProductionTracking]
GO
/****** Object:  Schema [app]    Script Date: 11/1/2015 9:12:33 PM ******/
CREATE SCHEMA [app]
GO
/****** Object:  Table [app].[Hours]    Script Date: 11/1/2015 9:12:33 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [app].[Hours](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Description] [varchar](50) NOT NULL,
	[From] [time](7) NOT NULL,
	[To] [time](7) NOT NULL,
	[StatusId] [int] NOT NULL,
 CONSTRAINT [PK_Hours] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]


GO

SET ANSI_PADDING OFF

GO

/****** Object:  Table [app].[ProductionLines]    Script Date: 11/1/2015 9:12:33 PM ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [app].[ProductionLines](

	[Id] [int] IDENTITY(1,1) NOT NULL,

	[Description] [varchar](50) NOT NULL,

	[StatusId] [int] NOT NULL,

 CONSTRAINT [PK_ProductionLines] PRIMARY KEY CLUSTERED

(

	[Id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]

) ON [PRIMARY]
```

```
GO

SET ANSI_PADDING OFF

GO

/****** Object:  Table [app].[ProductionPlanActual]    Script Date: 11/1/2015 9:12:33 PM
******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [app].[ProductionPlanActual](

	[Id] [bigint] IDENTITY(1,1) NOT NULL,

	[Date] [date] NOT NULL,

	[ProdLineNo] [int] NOT NULL,

	[Hour] [int] NOT NULL,

	[Amount] [int] NOT NULL,

	[CreatedDateTime] [datetime] NULL,

	[CreatedBy] [varchar](50) NULL,

	[ModifiedDateTime] [datetime] NULL,

	[ModifiedBy] [varchar](50) NULL,

 CONSTRAINT [PK_ProductionPlanActual] PRIMARY KEY CLUSTERED

(

	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]


GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [app].[ProductionPlanBudget]    Script Date: 11/1/2015 9:12:33 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [app].[ProductionPlanBudget](
	[Id] [bigint] IDENTITY(1,1) NOT NULL,
	[Date] [date] NOT NULL,
	[ProdLineNo] [int] NOT NULL,
	[Hour] [int] NOT NULL,
	[Amount] [int] NOT NULL,
	[CreatedDateTime] [datetime] NULL,
	[CreatedBy] [varchar](50) NULL,
	[ModifiedDateTime] [datetime] NULL,
	[ModifiedBy] [varchar](50) NULL,
 CONSTRAINT [PK_MonthlyPlanBudget] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]


GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [app].[Status]    Script Date: 11/1/2015 9:12:33 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [app].[Status](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[Description] [varchar](50) NOT NULL,
 CONSTRAINT [PK_Status] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]


GO
SET ANSI_PADDING OFF
GO
/****** Object: Table [dbo].[__MigrationHistory]    Script Date: 11/1/2015 9:12:33 PM
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[__MigrationHistory](
	[MigrationId] [nvarchar](150) NOT NULL,
	[ContextKey] [nvarchar](300) NOT NULL,
	[Model] [varbinary](max) NOT NULL,
	[ProductVersion] [nvarchar](32) NOT NULL,
 CONSTRAINT [PK_dbo.__MigrationHistory] PRIMARY KEY CLUSTERED
(
	[MigrationId] ASC,
	[ContextKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]


GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[AspNetRoles]    Script Date: 11/1/2015 9:12:33 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetRoles](
	[Id] [nvarchar](128) NOT NULL,
	[Name] [nvarchar](256) NOT NULL,
 CONSTRAINT [PK_dbo.AspNetRoles] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
GO
/****** Object:  Table [dbo].[AspNetUserClaims]    Script Date: 11/1/2015 9:12:33 PM
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUserClaims](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[UserId] [nvarchar](128) NOT NULL,
	[ClaimType] [nvarchar](max) NULL,
	[ClaimValue] [nvarchar](max) NULL,
 CONSTRAINT [PK_dbo.AspNetUserClaims] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/****** Object:  Table [dbo].[AspNetUserLogins]    Script Date: 11/1/2015 9:12:33 PM
******/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AspNetUserLogins](

	[LoginProvider] [nvarchar](128) NOT NULL,

	[ProviderKey] [nvarchar](128) NOT NULL,

	[UserId] [nvarchar](128) NOT NULL,

 CONSTRAINT [PK_dbo.AspNetUserLogins] PRIMARY KEY CLUSTERED

(

	[LoginProvider] ASC,

	[ProviderKey] ASC,

	[UserId] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]

) ON [PRIMARY]


GO

/****** Object:  Table [dbo].[AspNetUserRoles]    Script Date: 11/1/2015 9:12:33 PM ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[AspNetUserRoles](

	[UserId] [nvarchar](128) NOT NULL,
	[RoleId] [nvarchar](128) NOT NULL,
 CONSTRAINT [PK_dbo.AspNetUserRoles] PRIMARY KEY CLUSTERED
(
	[UserId] ASC,
	[RoleId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]


GO
/****** Object:  Table [dbo].[AspNetUsers]    Script Date: 11/1/2015 9:12:33 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUsers](
	[Id] [nvarchar](128) NOT NULL,
	[Email] [nvarchar](256) NULL,
	[EmailConfirmed] [bit] NOT NULL,
	[PasswordHash] [nvarchar](max) NULL,
	[SecurityStamp] [nvarchar](max) NULL,
	[PhoneNumber] [nvarchar](max) NULL,
	[PhoneNumberConfirmed] [bit] NOT NULL,
	[TwoFactorEnabled] [bit] NOT NULL,
	[LockoutEndDateUtc] [datetime] NULL,
	[LockoutEnabled] [bit] NOT NULL,
	[AccessFailedCount] [int] NOT NULL,
	[UserName] [nvarchar](256) NOT NULL,
 CONSTRAINT [PK_dbo.AspNetUsers] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]


GO
SET IDENTITY_INSERT [app].[ProductionLines] ON


GO
INSERT [app].[ProductionLines] ([Id], [Description], [StatusId]) VALUES (3, N'Line 1', 1)
GO
INSERT [app].[ProductionLines] ([Id], [Description], [StatusId]) VALUES (4, N'Line 2', 1)
GO
INSERT [app].[ProductionLines] ([Id], [Description], [StatusId]) VALUES (6, N'Line 3', 1)
GO
INSERT [app].[ProductionLines] ([Id], [Description], [StatusId]) VALUES (8, N'Line 4', 1)
```

```
GO

SET IDENTITY_INSERT [app].[ProductionLines] OFF

GO

SET IDENTITY_INSERT [app].[ProductionPlanActual] ON


GO

INSERT [app].[ProductionPlanActual] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (12,
CAST(N'2015-11-01' AS Date), 3, 12, 112, CAST(N'2015-11-01 16:22:41.540' AS DateTime),
N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanActual] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (13,
CAST(N'2015-11-01' AS Date), 4, 12, 112, CAST(N'2015-11-01 16:22:41.540' AS DateTime),
N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanActual] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (14,
CAST(N'2015-11-01' AS Date), 6, 12, 112, CAST(N'2015-11-01 16:22:41.540' AS DateTime),
N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanActual] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (15,
CAST(N'2015-11-01' AS Date), 8, 12, 112, CAST(N'2015-11-01 16:22:41.540' AS DateTime),
N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanActual] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (16,
CAST(N'2015-11-01' AS Date), 3, 12, 112, CAST(N'2015-11-01 16:22:41.540' AS DateTime),
N", NULL, NULL)

GO

INSERT [app].[ProductionPlanActual] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (17,
CAST(N'2015-11-01' AS Date), 4, 12, 112, CAST(N'2015-11-01 16:22:41.540' AS DateTime),
N", NULL, NULL)

GO

INSERT [app].[ProductionPlanActual] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (18,
CAST(N'2015-11-01' AS Date), 6, 12, 112, CAST(N'2015-11-01 16:22:41.540' AS DateTime),
N", NULL, NULL)

GO

INSERT [app].[ProductionPlanActual] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (19,
CAST(N'2015-11-01' AS Date), 8, 12, 112, CAST(N'2015-11-01 16:22:41.540' AS DateTime),
N", NULL, NULL)

GO

INSERT [app].[ProductionPlanActual] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (20,
CAST(N'2015-11-01' AS Date), 3, 12, 112, CAST(N'2015-11-01 16:22:41.540' AS DateTime),
N", NULL, NULL)

GO

INSERT [app].[ProductionPlanActual] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (21,
CAST(N'2015-11-01' AS Date), 4, 12, 112, CAST(N'2015-11-01 16:22:41.540' AS DateTime),
N", NULL, NULL)

GO
```

```
INSERT [app].[ProductionPlanActual] ([Id], [Date], [ProdLineNo], [Hour], [Amount], [CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (22, CAST(N'2015-11-01' AS Date), 6, 12, 112, CAST(N'2015-11-01 16:22:41.540' AS DateTime), N'', NULL, NULL)

GO

SET IDENTITY_INSERT [app].[ProductionPlanActual] OFF

GO

SET IDENTITY_INSERT [app].[ProductionPlanBudget] ON


GO

INSERT [app].[ProductionPlanBudget] ([Id], [Date], [ProdLineNo], [Hour], [Amount], [CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (23, CAST(N'2015-11-01' AS Date), 3, 12, 112, CAST(N'2015-11-01 16:22:12.783' AS DateTime), N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanBudget] ([Id], [Date], [ProdLineNo], [Hour], [Amount], [CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (24, CAST(N'2015-11-01' AS Date), 4, 12, 112, CAST(N'2015-11-01 16:22:12.783' AS DateTime), N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanBudget] ([Id], [Date], [ProdLineNo], [Hour], [Amount], [CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (25, CAST(N'2015-11-01' AS Date), 6, 12, 112, CAST(N'2015-11-01 16:22:12.783' AS DateTime), N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanBudget] ([Id], [Date], [ProdLineNo], [Hour], [Amount], [CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (26,
CAST(N'2015-11-01' AS Date), 8, 12, 112, CAST(N'2015-11-01 16:22:12.783' AS DateTime),
N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanBudget] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (27,
CAST(N'2015-11-01' AS Date), 3, 12, 112, CAST(N'2015-11-01 16:22:12.783' AS DateTime),
N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanBudget] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (28,
CAST(N'2015-11-01' AS Date), 4, 12, 112, CAST(N'2015-11-01 16:22:12.783' AS DateTime),
N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanBudget] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (29,
CAST(N'2015-11-01' AS Date), 6, 12, 112, CAST(N'2015-11-01 16:22:12.783' AS DateTime),
N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanBudget] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (30,
CAST(N'2015-11-01' AS Date), 8, 12, 112, CAST(N'2015-11-01 16:22:12.783' AS DateTime),
N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanBudget] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (31,
CAST(N'2015-11-01' AS Date), 3, 12, 112, CAST(N'2015-11-01 16:22:12.783' AS DateTime),
N'', NULL, NULL)

GO
```

```
INSERT [app].[ProductionPlanBudget] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (32,
CAST(N'2015-11-01' AS Date), 4, 12, 112, CAST(N'2015-11-01 16:22:12.783' AS DateTime),
N'', NULL, NULL)

GO

INSERT [app].[ProductionPlanBudget] ([Id], [Date], [ProdLineNo], [Hour], [Amount],
[CreatedDateTime], [CreatedBy], [ModifiedDateTime], [ModifiedBy]) VALUES (33,
CAST(N'2015-11-01' AS Date), 6, 12, 112, CAST(N'2015-11-01 16:22:12.783' AS DateTime),
N'', NULL, NULL)

GO

SET IDENTITY_INSERT [app].[ProductionPlanBudget] OFF

GO

SET IDENTITY_INSERT [app].[Status] ON


GO

INSERT [app].[Status] ([Id], [Description]) VALUES (1, N'Active')

GO

INSERT [app].[Status] ([Id], [Description]) VALUES (2, N'Inactive')

GO

SET IDENTITY_INSERT [app].[Status] OFF

GO

INSERT [dbo].[__MigrationHistory] ([MigrationId], [ContextKey], [Model],
[ProductVersion]) VALUES (N'201511010726091_InitialCreate',
N'MITProductionTracking.Models.ApplicationDbContext',
0x1F8B0800000000000400DD5CDB6EDC36107D2FD07F10F4D416CECA9726488D7502
676DB746E30BB29BA06F0157E2AE8548942A518E8DA25FD6877E527FA14389BAF1A
2CBAEBCBB0E0A145E697866383C2487A361FEFBE7DFF1DB07DF33EE7114BB013931
```

```
0F46FBA681891D382E599E98095DBC786DBE7DF3FD77E373C77F303EE572474C0E5A
92F8C4BCA3343CB6ACD8BEC33E8A47BE6B47411C2CE8C80E7C0B398175B8BFFF8B
75706061803001CB30C61F12425D1FA73FE0E72420360E6982BCABC0C15ECC9FC39B
698A6A5C231FC721B2F189797539BB8D0227B12958328B90FD054C1D65ED4CE3D473
11D834C5DEC234102101454CEEF8638CA7340AC8721AC203E4CD1E430C720BE4C598
F7E4B814EFDAA9FD43D629AB6C9843D9494C03BF27E0C111F79225365FC9D766E145
F0E339F89B3EB25EA7BE3C312F1D9C3EFA1078E00051E1F1C48B9830B8BB50711A87
D7988EF286A30CF22202B8AF41F4655445DC333AB7DB2B587538DA67FFED1993C4A
349844F084E6884BC3DE336997BAEFD3B7E9C055F3039393A982F8E5EBF7C859CA35
73FE3A397D59E425F41AEF6001E0163421C816D7851F4DF34AC7A3B4B6C5834ABB4
C9BC025C02D699C6157A788FC992DEC1D4397C6D1A17EE0376F2279C5C1F890BF309
1AD128819FD789E7A1B9878BF756A34EF6FF06AD872F5F0DA2F51ADDBBCB74E805
FD30712298571FB097BE8DEFDC309B5EB5F1FECCC52EA2C067BFEBFCCADE7E9E0
64964B3CE045A91198A9698D6AD1B5B25793B519A410D4FEB1C75F7A9CD2C95E9A
D14651D5A6526E42A363D1B727B9F566F67C69D86210C5E4A2DE69146C2356C5B230
10728A6922EE974D0954E04BAF92DAF8EE73E72BD0196C70E5A204659B8918F8B5EB
E0B808C88F4B6F916C531AC0ECE6F28BE6B301DFE1CC0F429B69308483BA5C80F9F5
CDBED5D40F075E2CFD95CD89CAEC18666F635B840360DA273C25AAD8DF73EB0BF
04093D27CE19A2F823B57340F673E6FADD010631E7D4B6711C5F0099B133092004CF0
12F093D3AEC0DC796AA6D8727130FB9BE3A3E1116D5CFB96819A3A825A4384523A6
8A559A4C7D1F2C5DD2CDD45C546F6A26D16A2A17EB6B2A03EB662997D41B9A0AB
4DA99490D16FDA523347CF897C2EE7EFCB7DEE6AD5B0B2A6E9CC20A897FC50447B
08C39B788521C917204BAAC1BDB0816D2E1634A9F7C6F4A357D425E32B4AA956643
BA080C3F1B52D8DD9F0DA999F0F8DE755854D2E150940B037C2779F579AB7DCE099
66D7A3AD4BAB969E59B590374D3E5348E03DB4D6781221DC6931975FB218633DA33
1B596FC4EC08740C88EEB22D0F9E40DF4C915437E40C7B9862E3D4CED2851314DBC
891DD081D727A1896EFA80AC3CA2C49DDB89F249DC0741CB146881D826298A92EA
1F2B47089ED86C86BF592D0B2E316C6FA5EE810DF9CE11013A6B0D5135D94AB9322
CC80428F30286D1E1A5B15C635135113B5EAC6BC2D842DC75DCA556C84932DB1B38
6973C7E7B1262367B6C03E46C76491703B409BE6D10949F55BA12403CB8EC1A418513
9386A03CA4DA0841EB1EDB0241EB2E797604CD8EA85DC75F38AFEE1A3DEB07E5C
D6FEB8DEEDA02376BFED8316A66B127B4A1D00247323DCFE6EC257EA08AC319D8
C9CF67310F75458A30F029A6F5944D19EF2AE350AB19442451136049B41650FE695002
9226540FE3F25C5EA3753C8AE8019BE7DD1A61F9DA2FC056382063573F915604F51F5
24572763A7D143D2BD82091BCD361A182A32084B878D53BDEC129BABCACEC982EB
```

```
1709F68B8D2313E180D0E6A895C354ECA3B33B897726AB67B491590F509C9D6F29210
3E69BC947766702F718EB63B491114F4080BD672517D0B1F68B2E5998E62B729DE8DA
DAC868A3F185B9A62ABF1150A43972C2BC557FC8931CD2AAF262FA6FD0B91FC0C
C3B263453D52616DA18906115A62E12DA8064B2FDC28A66788A23962799E89E34B62
CABD55B3FCE72AABDBA73C88F93E904BB3BFB3168D1FF46BBBAE1C9670B40BE8
ABCF629B34A1AE6082BAB9C1CAE2908722450E7F1278894FF4A196BE75F625AFDA3
E7B22238C2DC17E299492FC2605BCF541E83444F2F4187CB88A9866F521D343E81C9F
47A455D7EBA2543D4A9EB4AAA2E812595B1B425D70B3E2B0891164FF516B45789AB
9C6CB56AA00FC514F8C4AE583045679D71DB55E9C52C5ACBFE98E2854A054218557
3DACACD699D48CACBE58094FE351B544770D726549155D7EDB1D595163528556BC5
E015B61B3F8AE3BAAA20CA50AAC78DD1DBBAC491197D31DDECDB4E79A01B6B3
EC10BCDE7EA6C1789AB57198EDB0F2ADBF0A5479DC138B7FCD97C0F8F39DE495F
6243800AFB22CC87ABCD260E857A3DAF7F2FA62D4F8915F8F59FB085E5BF09B8A00
F478FDD8FBA41C918E84A248A1BD381A0A47C0313F8EB55FCA91CE67998869E46E8
4CDFE31A6D81F3181D1F44F6FE2B9982DEDB9C01522EE02C7342BFC300FF70F0E85
DB3CBB73B3C68A63C7531C6775D76BEA63B6811A2E728F22FB0E457245C51AB74F4
A5029597D491CFC7062FE95B63A4EF31EECAFF4F19E71197F24EE9F09BC98450936FE
962B4487A9C66F3E82EDE8DD89EE5EBDFCE373D674CFB88960C61C1BFB822F5719E
1FA8D8A5ED6644DD7B066E57B16CF7742D52E2C28518509B1FAFD84B94B07B99B90
5BF9838F1E7EEC6B9AF2FEC15A888A3B0643E10DE242DD1D8255B0B4F7071CF849D
3FB03FD3AABBE4FB08A69DABB042EE90F26DE24E8BE0CE52DB7B8D5284E479B58
92523FB75662AF5596B9EDBD492AD85E6BA2CB45D93DE0D628BC5E8119CFAC6679
B0DD5151923C18F636A9FDE475C8BB527A5C16856CB7E2789345C60D1F8DBEA9DA
E21DA8865354F76CBF8278D35CD3657477BC0CB35F9DF08E918DD77C6DBF1A78D36
4D3A579779C6CBD6A7E778C6BDBDA3FB7CCB4CE5BE8D62B78E56224CD9719552E
B8AD42374B9CC3097F1E0009B28832BB58A92E096B2A676D51588AE895EA6BD144C
5D2C491F44A12CD6AFBF5956FF88D9DE532CD6A35159C4DBAF9FADFA89BCB34EB
D6D4456EA3B6585999A8AAF76E59C79A4AA49E532D71AD272DA5EB6D316BE367F6
E7543A3C88536AB347F38DF8F9540A0FE29221A74E8FCA60F9732FEC9D957FA911F6
EFD85D9610ECDF6D24D8AEED9A85CC255904F9E62D58948B08199A2B4C91035BEA
6944DD05B229BC6639E6F466789AB7635F3AE6D8B92437090D130A5DC6FEDCAB25B
C5810D0A43F2D7FAEDB3CBE09D9AF78882E80992ECBCDDF907789EB3985DD178A9
C90068245173CA3CBC692B2CCEEF2B140BA0E484720EEBE22289A613FF4002CBE215
3748F57B10DE8F71E2F91FD5866007520ED035177FBF8CC45CB08F931C728DBC34FE0
B0E33FBCF91FB548365DB0540000, N'6.1.1-30610')
```

```
GO

INSERT [dbo].[AspNetUsers] ([Id], [Email], [EmailConfirmed], [PasswordHash],
[SecurityStamp], [PhoneNumber], [PhoneNumberConfirmed], [TwoFactorEnabled],
[LockoutEndDateUtc], [LockoutEnabled], [AccessFailedCount], [UserName]) VALUES
(N'faa7fb9a-0897-44dd-952e-fa846d0df03e', N'dln0925@gmail.com', 0,
N'AMRsHAZpEPix8Z+6l0JfYGj75HgTz143BvD+uAgPqLD0IT4w+7TQjCk4F0NcF0Xo3w
==', N'046a58da-2893-4089-85cb-62391813f327', NULL, 0, 0, NULL, 1, 0,
N'dln0925@gmail.com')

GO

ALTER TABLE [app].[Hours]  WITH CHECK ADD  CONSTRAINT [FK_Hours_Status]
FOREIGN KEY([StatusId])

REFERENCES [app].[Status] ([Id])

GO

ALTER TABLE [app].[Hours] CHECK CONSTRAINT [FK_Hours_Status]

GO

ALTER TABLE [app].[ProductionLines]  WITH CHECK ADD  CONSTRAINT
[FK_ProductionLines_Status] FOREIGN KEY([StatusId])

REFERENCES [app].[Status] ([Id])

GO

ALTER TABLE [app].[ProductionLines] CHECK CONSTRAINT
[FK_ProductionLines_Status]

GO

ALTER TABLE [dbo].[AspNetUserClaims]  WITH CHECK ADD  CONSTRAINT
[FK_dbo.AspNetUserClaims_dbo.AspNetUsers_UserId] FOREIGN KEY([UserId])

REFERENCES [dbo].[AspNetUsers] ([Id])

ON DELETE CASCADE

GO
```

```
ALTER TABLE [dbo].[AspNetUserClaims] CHECK CONSTRAINT
[FK_dbo.AspNetUserClaims_dbo.AspNetUsers_UserId]

GO

ALTER TABLE [dbo].[AspNetUserLogins] WITH CHECK ADD CONSTRAINT
[FK_dbo.AspNetUserLogins_dbo.AspNetUsers_UserId] FOREIGN KEY([UserId])

REFERENCES [dbo].[AspNetUsers] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserLogins] CHECK CONSTRAINT
[FK_dbo.AspNetUserLogins_dbo.AspNetUsers_UserId]

GO

ALTER TABLE [dbo].[AspNetUserRoles] WITH CHECK ADD CONSTRAINT
[FK_dbo.AspNetUserRoles_dbo.AspNetRoles_RoleId] FOREIGN KEY([RoleId])

REFERENCES [dbo].[AspNetRoles] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserRoles] CHECK CONSTRAINT
[FK_dbo.AspNetUserRoles_dbo.AspNetRoles_RoleId]

GO

ALTER TABLE [dbo].[AspNetUserRoles] WITH CHECK ADD CONSTRAINT
[FK_dbo.AspNetUserRoles_dbo.AspNetUsers_UserId] FOREIGN KEY([UserId])

REFERENCES [dbo].[AspNetUsers] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserRoles] CHECK CONSTRAINT
[FK_dbo.AspNetUserRoles_dbo.AspNetUsers_UserId]
```

GO

Appendix C: Test Cases

ID	Test Case	Steps	Expected Results	Actual Results	Status
5.0	**Add Actuals**				
	Add Actual details	Upload the Actual with correct upload format correct details	details upload to the system	Same	Passed
		Upload the Actual with correct upload format wrong details	warning message should appear	Same	Passed
		Upload the Actual with In-correct upload format correct details	warning message should appear	Same	Passed
		Upload the Actual with In-correct upload format in-correct details	warning message should appear	Same	Passed
6.0	**Add Lines**				
	Add new lines	Add a line with numeric value as the line name	line gets added	Same	Passed
		Add a line with non-numeric value as the line name	line gets added	Same	Passed
7.0	**Edit Budget**				
	Edit production budget	Click the edit button in the table and change the value and hit save button	New value get added	Same	Passed
		Click the edit button in the table and change the value to a	Warning message shows up	Same	Passed

		non-numeric value and hit the save button			
8.0	**Edit Actuals**				
	Edit production Actuals	Click the edit button in the table and change the value and hit save button	New value get added	Same	Passed
		Click the edit button in the table and change the value to a non-numeric value and hit the save button	Warning message shows up	Same	Passed
9.0	**Edit Lines**				
	Edit lines	Click the edit button change the status ID to 2	Line status changed to disable	Same	Passed
		Click the edit button change the status ID to 1	Line status changed to enable	Same	Passed
		Click the edit button change the status ID to 3	Warning message shows up	Same	Passed
10.0	**Add Feedback**				
	Add line feedback	Select a line and add feedback and save	Details get saved to the system	same	Passed
		Click save button without adding details	Warning message shows up	same	Passed
11.0	**View Lines**				

	View the active lines	Click the edit button	Go to edit form	Same	Passed