TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

Diploma in Software Engineering

Programme: DSFY1S1 (Group: 2)

Assignment

AMSE1003 SOFTWARE ENGINEERING

Name (Block Letters)	Registration No.	Signature	Marks
1.Low Jia Ming	24SMD09939	Ming	
2.Vincent Chiew Jia Cheng	24SMD02394	Vincent	
3.Ian Chin Kar Le	24SMD11429	Ian	
4.Yong Hao Cheng	24SMD01385	Yong	
5.Fong Jia Ming	24SMD07100	Fong	

Lecturer's Name: SURAYAINI BINTI BASRI

Date of Submission: 9/28/2024

TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

Plagiarism Statement and Guideline for Late Submission of Coursework

Read, complete, and sign this statement to be submitted with the written report.

We confirm that the submitted works are all our own work and are in our own words.

Name (Block Letters)	Registration No.	Signature	Date
1.Low Jia Ming	24SMD09939	Ming	9/28/2024
2.Vincent Chiew Jia Cheng	24SMD02394	Vincent	9/28/2024
3.Ian Chin Kar Le	24SMD11429	Ian	9/28/2024
4.Yong Hao Cheng	24SMD01385	Yong	9/28/2024
5.Fong Jia Ming	24SMD07100	Fong	9/28/2024

TABLE OF CONTENTS

Company name: Jia Hua Food Court

Location: Grand Industrial Centre, 89500 Penampang, Sabah

What they do: Restaurant/Food court selling buns, cakes and drinks. They also rent empty lots for vendors to sell their own food.

Part 1:

Major problems of the manual process:

1. Jia Hua wants the staff to handle orders manually for their own stall, but can mess up the ordering process because the writing can be misinterpreted by others or the staff can take the customer's orders incorrectly. This can cause mistakes such as taking the wrong food to the wrong table, thus leaving customers unsatisfied and leaving bad reviews causing reputation damage.
2. Jia Hua wants to record the vendors' information. Records of the company are written and kept physically in books. This means important documents and data regarding their patrons are easily stolen, which could spell the downfall of their business and face serious charges due to failure to keep said information confidential. Furthermore, there are no backups of destroyed records.
3. Jia Hua's manual leave management is prone to errors such as miscalculations, incorrect recordings of leave days, and miscommunication, which can lead to human resources mismanagement, payroll discrepancies and employee dissatisfaction.
4. Restocking supplies manually risks losing or misplacing supplies. As the supplies needed are ordered from various suppliers, poor management will lead to disasters such as overbuying supplies. These problems will give the management a hard time managing the supplies. Poor management can cause a delay in payment or break the budget of the company.
5. Affairs of the entire food court are problematic to handle without a proper system as there are many challenges that need to be addressed throughout the year. When people want to rent slots for vendoring, the company might have issues with slot allocation for the renting as information regarding available slots is mixed up easily if handled manually through paper and pen.

Software Quality Attributes

1. Acceptability

   By investing in acceptable software the restaurant staff members can minimize the amount of errors and inaccuracies during rush hour and normal restaurant operations due to how usable the software system is. It can also improve the service quality and customer satisfaction since the system will be more user friendly towards young and old customers.

2. Efficiency

   Time management will be more efficient. With this, the delay between each order made by a customer will be minimal, smoothing staff services towards customers. The system would also handle employee affairs and schedule with more efficiency and while minimizing use of space for the system.

3. Maintainability

   With the systematic software, it is easy to maintain changes in the company. Such as adding or removing selling goods and adjusting in goods prices in the system. It can also be flexible when it comes to employees' working schedule.

4. Dependability and Security

   The software system will be dependable with its reliability to prevent physical and economical damage in the event of system failure. The system will also be able to protect itself from cyberattacks such as viruses or hacking attempts by malicious users in order to access or damage the systems. For instance, hackers cannot steal or alter business data in the system.

Software Process Model

Prototyping Model - Evolutionary Prototyping is the software process model proposed for our system.

Software Process Model is a framework used to structure, plan, and control flow of work required to develop a software.It comprised of many specific activities and tasks.The goal is to provide guidance for systematically coordinating and controlling tasks that must be performed to achieve the end product objectives.In our software development, we use the model of evolutionary prototyping as a basis to achieve our goal. The evolutionary prototyping model is a structure where developers produce a firsthand prototype for showcasing to the client in question. This allows the client to gain a concrete impression of the system capabilities as well as experimenting with the requirements. This specific model is specifically aimed at clients that are unsure of or have yet to fully decide regarding requirements of the system as the prototype is not the final product.

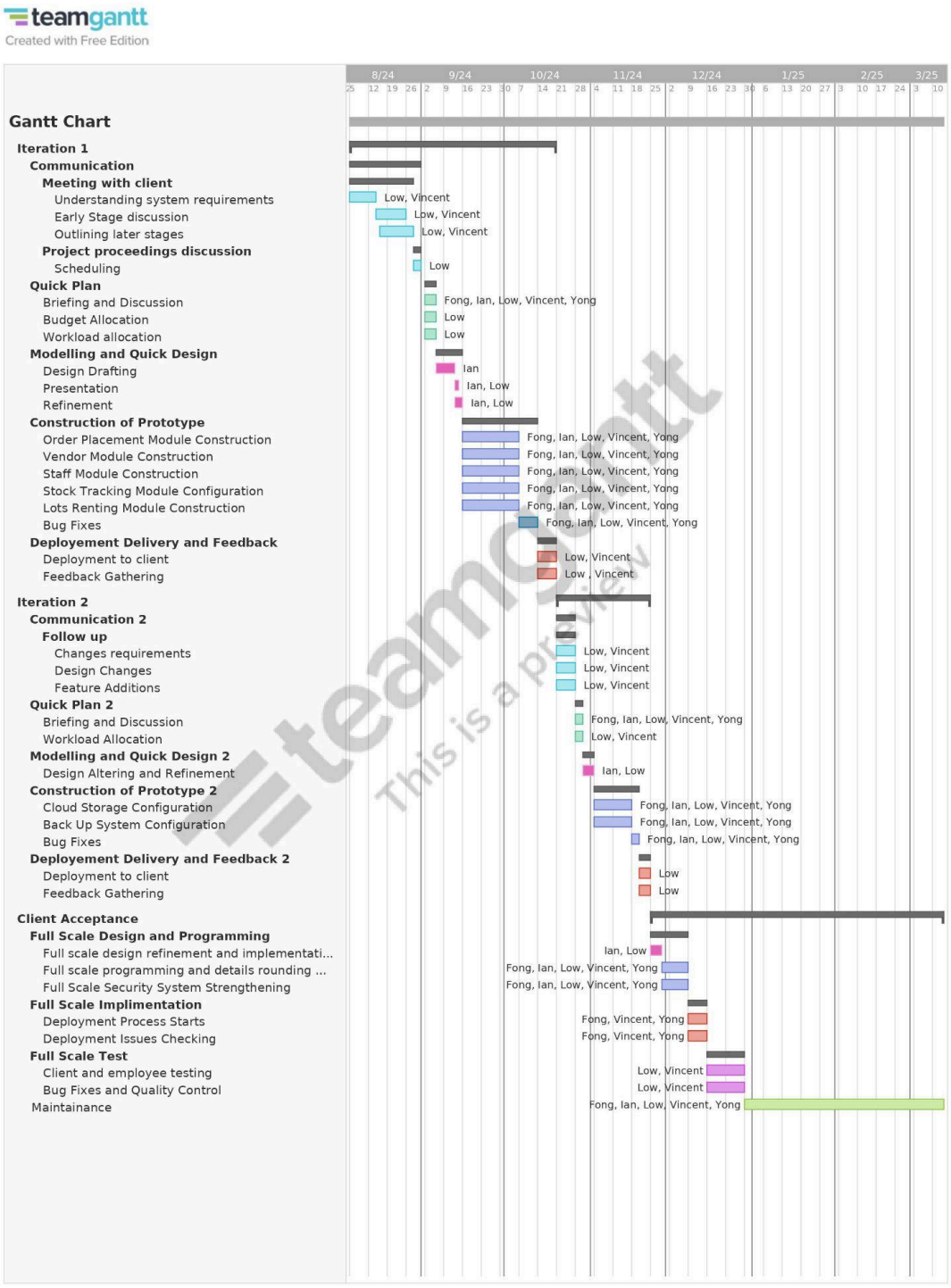
Since our client finds it difficult to express their real requirements, we proposed the use of the prototyping model as our software process model. This is due to the fact that the prototyping model allows the customer to experiment with requirements to confirm what features they want in the system at a rapid software delivery and utilization pace.

Of the two types of prototyping model, we will use the prototyping evolutionary.This approach allows the restaurant to continuously collect and analyze feedback from customers. The feedback enables the restaurant to make data-driven decisions and improvements to meet customer needs and preferences.Prototyping evolution will also help with engaging customer's interest and ensure effectiveness before full-scale Implementation. With that, the prototype can be further modified to suit the client's future demand through the evolutionary prototype model since the client likely wants it to work.

Part 2:

Project Plan and Schedule.

Gantt Chart



Duration

<table>
<tr><td rowspan="13">Iteration 1

75 Days</td><td colspan="3">Communication 25 Days</td></tr>
<tr><td rowspan="3">Meeting Client

23 Days</td><td colspan="2">Understanding System Requirements 9 Days</td></tr>
<tr><td colspan="2">Early stage discussion 10 Days</td></tr>
<tr><td colspan="2">Outlining later stages 13 Days</td></tr>
<tr><td>Project Proceedings Discussion 2 Days</td><td colspan="2">Scheduling 2 Days</td></tr>
<tr><td colspan="3">Quick Plan 3 Days</td></tr>
<tr><td colspan="2">Briefing and Discussion</td><td rowspan="3">3 Days</td></tr>
<tr><td colspan="2">Budget Allocation</td></tr>
<tr><td colspan="2">Workload Allocation</td></tr>
<tr><td colspan="3">Modeling and Quick Design 9 Days</td></tr>
<tr><td colspan="2">Design Drafting</td><td>7 Days</td></tr>
<tr><td colspan="2">Presentation</td><td>1 Days</td></tr>
<tr><td colspan="2">Refinement</td><td>2 Days (Overlap 1 Day)</td></tr>
<tr><td colspan="3">Construction of Prototype 26 Days</td></tr>
<tr><td colspan="2">Order Placement Module Construction</td><td rowspan="4">19 Days</td></tr>
<tr><td colspan="2">Vendor Module Construction</td></tr>
<tr><td colspan="2">Staff Module Construction</td></tr>
<tr><td colspan="2">Stock Tracking Module Construction</td></tr>
</table>

	Lots Renting Module Construction	
	Bug Fixes	5 Days
	Deployment Delivery and Feedback 5 Days	
	Deployment to client	5 Days
	Feedback Gathering	
Iteration 2 33 Days	**Communication2 5 Days**	
	Follow Up 5 Days	Changes requirements 5 Days
		Design Changes 5 Days
		Feature Additions 5 Days
	Quick Plan2 2 Days	
	Briefing and Discussion	2 Days
	Workload Allocation	
	Modeling and Quick Design2 3 Days	
	Design Altering and Refinement	3 Days
	Construction of Prototype2 16 Days	

	Cloud Storage Configuration	12 Days
	Back Up System Configuration	
	Bug Fixes	2 Days
	Deployment Delivery and Feedback 3 Days	
	Deployment to client	3 Days
	Feedback Gathering	

<table>
<tr><td rowspan="11">Client Acceptance 127 Days</td><td colspan="2">Full Scale Design and Programming 12 Days</td></tr>
<tr><td>Full scale refinement and implementation</td><td>3 Days</td></tr>
<tr><td>Full scale programming and details rounding up</td><td rowspan="2">9 Days</td></tr>
<tr><td>Full scale system security strengthening</td></tr>
<tr><td colspan="2">Full Scale Implementation 5 Days</td></tr>
<tr><td>Deployment Process Starts</td><td rowspan="2">5 Days</td></tr>
<tr><td>Deployment Issues Checking</td></tr>
<tr><td colspan="2">Full Scale Test 12 Days</td></tr>
<tr><td>Client and employee testing</td><td rowspan="2">12 Days</td></tr>
<tr><td>Bug Fixes and Quality Control</td></tr>
<tr><td>Maintenance</td><td colspan="2">92 Days</td></tr>
</table>

Software Requirements Specification.

Functional Requirements:

Module 1 - Ordering Module:

1.1 The system shall allow the staff to select the menu items the customer wants to order and record the customer's table number.

1.2 The system shall allow staff to make changes to the order.

1.3 The system shall display the customer's orders to the kitchen staff.

Module 2 - Vendor Module:

2.1 The system shall be able to store data such as vendors' information.

2.2 the system shall allow staff to edit vendor information

2.3 The system shall have a searching function that allow quick access to vendors' information

Module 3 - Staff Module:

3.1 The system shall allow the employees to view their leave history and display remaining leave days for each employee

3.2 The system shall allow employees to submit leave requests into the system

3.3 The system shall allow staff to check their leave application progress

Module 4 - Supply Tracking Module:

4.1 The system shall track the inventory levels of supplies daily.

4.2 The system shall notify management when supplies levels are less than 20% of the existing supplies.

4.3 The system shall generate detailed reports on supply orders and its payments to help stay within budget.

Module 5 - Lots Renting Module:

5.1 The system shall display available and unavailable slots for vendors to rent.

5.2 The system shall allow the management to check the vendors' rent status.

5.3 The system shall display information regarding the renter of each vendor.

Non-functional Requirements:

1.1 The system response time shall not exceed 3 seconds.

1.2 The system shall backup records daily.

1.3 The system shall not use more than 500MB of storage space.

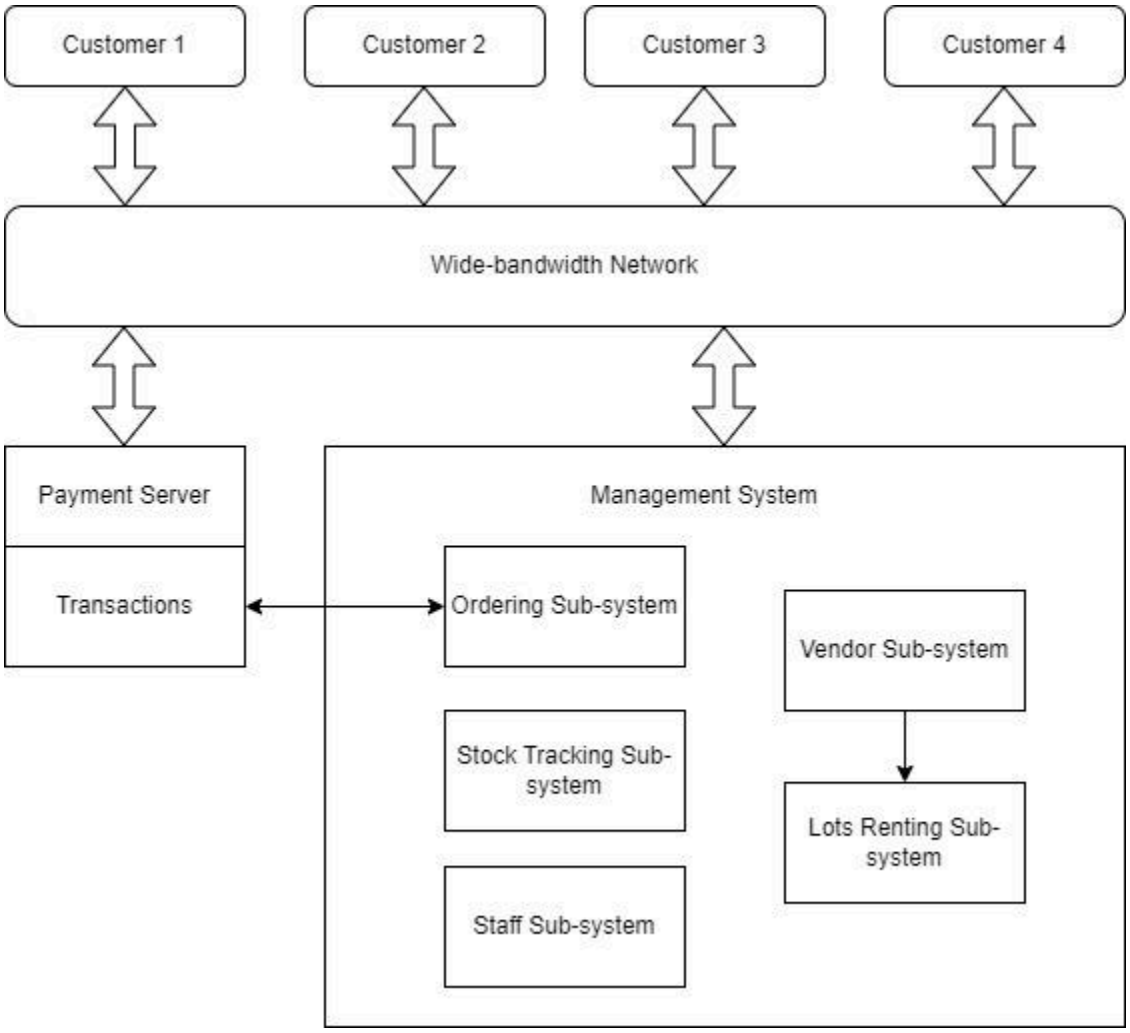
1.4 The system shall have an uptime of at least 99.9% to ensure continuous operation, minimizing the risk of losing or misplacing supply orders due to system downtime.

1.5 The system shall not disclose any personal information to unauthorized personnel.

1.6 The system shall display information in a clean, clear and organized manner.

1.7 The system shall be able to protect the data with strong passwords

Architectural Design



Explanation:

Client Based Model is a distributed system model which shows how data and processing is distributed across a range of components.It consists of a set of stand-alone servers which provide specific services such as printing, data management, etc.

Justification:

We use this system because it utilizes cheaper hardware. Its network system is an incredibly effective system and the distribution of data is straightforward, simplifying the process.

Part 3:

Test Cases

Test Case Name	1.1.1 Select valid menu items but record invalid table number	Test Case Description	To check the ordering function by using valid menu item but invalid table number
Pre-conditions:		**Test Data:**	
1	The staff has valid authentication	1	order = "Bun Kahwin"
2		2	table number = 2190876

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click the "Start New Order" button	The ordering menu page is displayed.		
2	Select the items	The customer's order is selected.		
3	Type in the invalid table number	The invalid table number message is displayed. Table number is prompted for input until valid table number.		
4	Click the "Confirm" button	The customer's order is confirmed.		

Test Case Name	1.2.1 Change the order to a valid menu item	Test Case Description	To test the order changing function by changing the order to a valid menu item
Pre-conditions:		**Test Data:**	
1	The staff has valid authentication	1	new order = "Roti Kahwin"
2	The table number is valid	2	
3	The order has yet to be confirmed	3	

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click the "Edit Order" button	The order editing page is displayed.		
2	Select the item to be edited	The item to be edited is selected and menu items are displayed to change the item to.		
3	Select the new item	The new item is selected. The updated list of items in the order is displayed.		
4	Click the "Confirm" button	The customer's order is confirmed.		

Test Case Name	1.3.1 Display valid customer's order and its order number to the kitchen	Test Case Description	To test the order displaying function to the kitchen
Pre-conditions:		**Test Data:**	
1	The user has valid authentication	1	order number = 3
2	The order is valid and confirmed	2	

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click the "Show Order" button	The list of customers' orders is displayed.		
2	Choose the order number to view	The order that is intended to be viewed is selected. The list of ordered items and their quantity for that order number are displayed.		
3	Click the "Start Preparing" button	The status of the order is changed to being prepared and is displayed.		

Test Case Name	2.1.1 Store valid data such as vendors' information.	Test Case Description	To test the function to store data of sales and vendors' information
Pre-conditions:		**Test Data:**	
1	Staff have correct authentication	1	Vendor_name = "Ian"

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click 'Settings' button	Settings page is displayed.		
2	Click 'Back Up Centre' button	Back up center is displayed.		
3	Type in the valid vendors' information	Vendors' information is entered		
4	Click the ' Back Up ' button	The vendors' information is saved		

Test Case Name	2.2.1 Staff edit vendor information	Test Case Description	To test editing function of vendors' information
Pre-conditions:		**Test Data:**	
1	Staff have valid authentication	1	Edit new vendors' name = Ian2

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click "Vendors" button	Section of Vendors is displayed		
2	Click "Vendors Information" button	Valid vendors' personal information is displayed		
3	Click the vendor's name	Display "Edit" button		
4	Click the "Edit" button	The vendor's name become editable		
5	Edit the vendor's name to "Ian2"	"Ian2" is displayed as new the new vendor's name		

Test Case Name	2.3.1 Staff search for required data	Test Case Description	To test search function of vendor's name
Pre-conditions:		**Test Data:**	
1	A vendor name is "Ian2"	1	Search : "Ian2"

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click "Vendors" button	Section of Vendors is displayed		
2	Click "Vendors Information" button	Valid vendors' personal information is displayed		
.	Fill in "Ian2" in the search bar	Vendor Ian2's personal information is displayed.		

Test Case Name	3.1.1 shows valid leave history of the staff members	Test Case Description	To confirm the system will show the correct and valid leave history.
Pre-conditions:		**Test Data:**	
1	Staff must be registered in the system	1	

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click "Application" button	Application section will be displayed		
2	Click "My Leave" button	Staff leave status section will be displayed		
3	Click "View History" button	Leave history will be displayed		

Test Case Name	3.2.1 Leave request application from the staff members	Test Case Description	To show that the system will submit leave requests for the staff
Pre-conditions:		**Test Data:**	
1	Staff must be registered in the system	1	

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click "Application" button	Application section will be displayed		
2	Click "Apply leave" button	Window to fill in the leave application detail is displayed.		
3	Fill in the details	Details are entered as typed		
4	Click "Submit" button	Status of pending leave application is displayed.		

Test Case Name	3.3.1 displaying leave application progress	Test Case Description	To show the valid progression of the leave request for the staff members
Pre-conditions:		**Test Data:**	
1	Have staff already registered a leave	1	

Step #	Step Detail	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click “Application” button	Application section will be displayed		
2	Click “My Leave” button	Staff leave status section will be displayed		
3	Click ”Check My Leave Request Progression “ button	Leave request application progress is displayed		

Test Case Name	4.1.1 Track the inventory levels of supplies daily.	Test Case Description	Making sure inventory has enough supplies for daily business.
Pre-conditions:		**Test Data:**	
1	The inventory must not be empty	1	

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click "Show Inventory" button	Inventory Selection is displayed.		
2	Click "Show Supplies" button	Current supplies and amount of supplies used is displayed.		

Test Case Name	4.2.1 Notify if supplies less than 20%	Test Case Description	Will notify management to order a certain supply that has a level lower than 20%.
Pre-conditions:		**Test Data:**	
1	The Inventory must not be empty	1	Cake flour lower than 19%
2	A supply of its level is less than 20%	2	Sugar lower than 9%

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click "Show Inventory" button	The Inventory Selection is displayed.		
2	Click "Inventory Notification" button	Inventory notification about supplies level lower than 20% is displayed.		

Test Case Name	4.3.1 Generate Reports for Supplies order	Test Case Description	Will generate detailed reports about all the supplies ordered by the user and whether the order has been paid and give the user remainder of payment due.
Pre-conditions:		**Test Data:**	
1	The order must be confirmed	1	Ordered 3 bags of sugar
2		2	Payment for 10 bags of cake flour
3		3	Payment due for 5 crates of 100 plus

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click "Show Inventory" button	Inventory Selection is displayed.		
2	Click "Inventory Reports" button	Detailed reports of the Inventory are displayed.		

Test Case Name	5.1.1 Display valid slots availability for rent	Test Case Description	To verify if system will display valid data accordingly
Pre-conditions:		**Test Data:**	
1	Staff have correct authentication	1	
2	All slots and vendors are registered in the database	2	

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click “Vendors” button	Section of Vendors is displayed		
2	Click “Slots” button	Valid slots availability is displayed		

Test Case Name	5.2.1 Display valid rent status of vendors	Test Case Description	To verify if system will display valid data accordingly
Pre-conditions:		**Test Data:**	
1	Staff have correct authentication	1	
2	All slots and vendors are registered in the database	2	

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click “Vendors” button	Section of Vendors is displayed		
2	Click “Rent Status” button	Valid vendors’ rent status is displayed		

Test Case Name	5.3.1 Display personal information of vendors	Test Case Description	To verify if system will display valid data accordingly
Pre-conditions:		**Test Data:**	
1	Staff have correct authentication	1	
2	All slots and vendors are registered in the database	2	

Step #	Step Details	Expected Results	Actual Results	Remarks (Pass / Fail / Not executed / Suspended)
1	Click “Vendors” button	Section of Vendors is displayed		
2	Click “Vendors Information” button	Valid vendors’ personal information is displayed		

Software Configuration Management:

https://github.com/LowJM/UniProgramming.git

In short, git has several main functions that are vital in helping us keep track of every change, stage changed, commit, push and history viewing. Stage change allows us to select a designated file and check and display changes in the file along with the original if the texts are edited instead of added.Commit function acts as a “screenshot” of the version and keeps it in the database while recording who is the author of changes should we ever need to revert back.Push function is just cloning or replace this file to connected repository and display the changes given the connected repository have such function while history viewing allows us to view past changes and done by who as well as resetting to that previous versions if required.This can be done through Visualise All Branch History or Visualise (your current branch) History.Git is also capable of creating branches to store even more variation and types of files or completely different files if needed.

Reference Section:

1. Priyanka, April 24, 2024, What Are The Software Quality Attributes?, testsigma.

   Available from: https://testsigma.com/blog/software-quality-attributes/

2. Maryam Sulemani, May 27, 2024, What is a software process model? Top 7 models explained,educative.

   Available from: https://www.educative.io/blog/software-process-model-types

3. 02DCE,12 Mar, 2024,Prototyping Model – Software Engineering,GeeksforGeeks.

   Available from: https://www.geeksforgeeks.org/software-engineering-prototyping-model/

4. Sarah Lewis,June 2023,What is the Prototyping Model, Techtarget.

   Available from: https://www.techtarget.com/searchcio/definition/Prototyping-Model

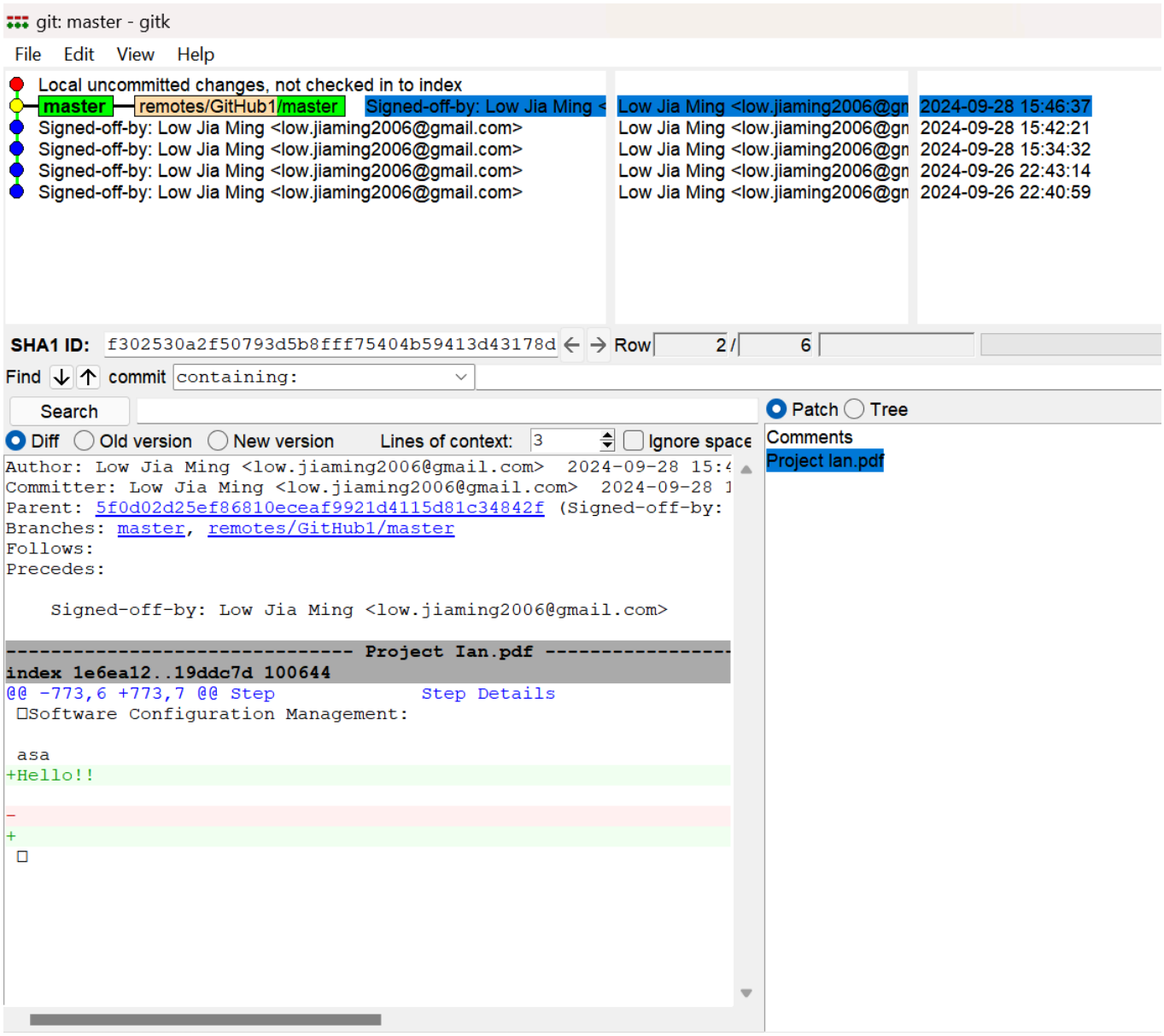
5. Lucas Bennett ,August 13, 2024,Prototype Model in Software Engineering, Guru99.

   Available from: https://www.guru99.com/software-engineering-prototyping-model.html

6. Kasper Joergensen, July 2023, Prototyping Gantt kort [classic], Creately.

   Available from:https://creately.com/diagram/example/ia5b3ufw1/prototyping-gantt-kort-classic

Appendices:


git: master - gitk
File Edit View Help
Local uncommitted changes, not checked in to index
master remotes/GitHub1/master Signed-off-by: Low Jia Ming <
Low Jia Ming <low.jiaming2006@gn 2024-09-28 15:46:37
Signed-off-by: Low Jia Ming <low.jiaming2006@gmail.com>
Low Jia Ming <low.jiaming2006@gn 2024-09-28 15:42:21
Signed-off-by: Low Jia Ming <low.jiaming2006@gmail.com>
Low Jia Ming <low.jiaming2006@gn 2024-09-28 15:34:32
Signed-off-by: Low Jia Ming <low.jiaming2006@gmail.com>
Low Jia Ming <low.jiaming2006@gn 2024-09-26 22:43:14
Signed-off-by: Low Jia Ming <low.jiaming2006@gmail.com>
Low Jia Ming <low.jiaming2006@gn 2024-09-26 22:40:59
SHA1 ID: f302530a2f50793d5b8fff75404b59413d43178d
Row 2 / 6
Find commit containing:
Search
Patch Tree
Diff Old version New version Lines of context: 3 Ignore space
Comments
Project Ian.pdf
Author: Low Jia Ming <low.jiaming2006@gmail.com> 2024-09-28 15:4
Committer: Low Jia Ming <low.jiaming2006@gmail.com> 2024-09-28 1
Parent: 5f0d02d25ef86810eceaf9921d4115d81c34842f (Signed-off-by:
Branches: master, remotes/GitHub1/master
Follows:
Precedes:
Signed-off-by: Low Jia Ming <low.jiaming2006@gmail.com>
---------- Project Ian.pdf ----------
index 1e6ea12..19ddc7d 100644
@@ -773,6 +773,7 @@ Step Step Details
Software Configuration Management:
asa
+Hello!!
-
+