IMPLEMENTATION rEPORT

Liu liu Megatech

Team 6

Contents

[Introduction 3](#_Toc511435654)

[Compilation 3](#_Toc511435655)

[Run Time 4](#_Toc511435656)

[Testing 5](#_Toc511435657)

[Unit Testing 5](#_Toc511435658)

[Upgrade Customer Account 5](#_Toc511435659)

[Print Late Payment Reminder 6](#_Toc511435660)

[Login 7](#_Toc511435661)

[Generate Reports 8](#_Toc511435662)

[Update Job Status 9](#_Toc511435663)

[Record Payment 10](#_Toc511435664)

[Generate 2nd Letter 12](#_Toc511435665)

[Add User 13](#_Toc511435666)

[Automatic Backup 14](#_Toc511435667)

[Update Existing Task: 15](#_Toc511435668)

[Non-Functional Testing 17](#_Toc511435669)

# Introduction

This is the implementation report for the BAPERS Software which covers the major tasks involved in the implementation process for the BAPERS system and diagrams to portray how the system would function. This provides an overview of the plan undertaken to ensure that the software was implemented correctly, and the required resources needed to support this process where thoroughly searched and used efficiently and effectively.

# Compilation

Description of how the system was compiled

# Run Time

Component diagram goes here

# Testing

## Unit Testing

We have made the following Test Cases based on our use case specs to test our BAPERS system

## Upgrade Customer Account

**Main Flow:**

|  |  |
| --- | --- |
| **Use case ID:** 13 | **Use case name:** Upgrade Customer Account |
| **Test number:** 1 | |
| **Objective:** Test the main flow | |
| **Set up:**   1. Create a customer called Barry Smith using the use case **Create A customer account (need to check)**. This customer should be applicable for the different discounts. 2. The office manager must be logged in to the system and have access to the customer list (this also allows him/her to manage the customer accounts), therefore, he will have access to Barry Smiths account details and can edit them at his discretion. | |
| **Expected results:**  1. The new “Valued” status will be assigned to Barry Smith.  2. The database will be updated with the new value for Barry Smith. | |
| **Test:**   1. Office Manager logs in to the system, accesses the customer list, clicks on Barry Smith. 2. Office Manager then clicks on the GUI to upgrade Barry Smith to the “Valued” status. | |
| **Test record:** The database is updated with the new Barry Smith “valued” status. | |
| **Date:xx/xx/xx** | **Tester:xxxxxxxxx** |
| **Result:xxxxx** | |
| **Date:** | **Tester:** |
| **Result:** | |

**Alternative Flow:**

|  |  |
| --- | --- |
| **Use case ID:** 13 | **Use case name:** Upgrade Customer Account |
| **Test number:** 2 | |
| **Objective:** Test the alternative flow where the process of making a customer “valued” cannot happen. | |
| **Set up:**   1. Create a customer called Joe Bloggs using the use case **Create A customer account (need to check)**. The customer account of Joe Bloggs should already have the status of a “valued” customer account. 2. The office manager must be logged in to the system and have access to the customer list (this also allows him/her to manage the customer accounts), therefore, he will have access to Joe Bloggs account details and can edit them at his discretion. | |
| **Expected results:**   1. The Office Manager will be prompted that the account of Joe Bloggs has already been upgraded to a valued customer account and cannot be further upgraded. | |
| **Test:**   1. Office Manager logs in to the system, accesses the customer list, clicks on Joe Bloggs. 2. Office Manager then clicks on the GUI to upgrade Joe Bloggs to the “Valued” status. | |
| **Test record:** The Office manager gets prompted Invalid Account Upgraded as Joe Bloggs is an existing valued customer. | |
| **Date:xx/xx/xx** | **Tester:xxxxxxxxx** |
| **Result:xxxxx** | |
| **Date:** | **Tester:** |
| **Result:** | |

## Print Late Payment Reminder

**Main Flow:**

|  |  |
| --- | --- |
| **Use case ID:** 7 | **Use case name:** Print Late Payment Reminder |
| **Test number:** 1 | |
| **Objective:** To test the main flow | |
| **Set up:**   1. Create a customer called John Doe using the use case **Create A customer account (need to check)**. 2. Set a job payment deadline for John Doe. 3. The office manager must be logged in to the system and accesses the customer list and sees that the payment deadline has passed at deadline day + 1 day and no payment has been made. | |
| **Expected results:**   1. A GUI pops up asking to print a late payment letter for John Doe. 2. Once the GUI is clicked a late payment is printed. | |
| **Test:**   1. Office Manager logs in to the system, accesses the customer list and sees that John Doe has missed his payment deadline. 2. GUI appears which the Office Manger clicks. | |
| **Test record:** A late payment letter is printed for John Doe after the GUI is clicked. | |
| **Date:xx/xxx/xxx** | **Tester:xxxxxxx** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** | |

**Alternative Flow:**

|  |  |
| --- | --- |
| **Use case ID:** 7 | **Use case name:** Print Late Payment Reminder |
| **Test number:** 2 | |
| **Objective:** To test the alternative flow where the Office Manager is unable to print the letters required due to a print error. | |
| **Set up:**   1. Create a customer called John Doe using the use case **Create A customer account (need to check)**. 2. Set a job payment deadline for John Doe. 3. The office manager must be logged in to the system and accesses the customer list and sees that the payment deadline has passed at deadline day + 1 day and no payment has been made. 4. An error with the printer not allowing any prints to be made. | |
| **Expected results:**   1. Late payment letter is not printed. | |
| **Test:**   1. Office Manager logs in to the system, accesses the customer list and sees that John Doe has missed his payment deadline. 2. GUI appears which the Office Manger clicks. | |
| **Test record:** A late payment letter is not printed for John Doe due to a printer error after the GUI is clicked. | |
| **Date:xx/xxx/xxx** | **Tester:xxxxxxx** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** | |

## Login

**Main Flow:**

|  |  |
| --- | --- |
| **Use case ID:** 5 | **Use case name:** Login |
| **Test number:** 1 | |
| **Objective:** To test the main flow | |
| **Set up:**   1. An account must be made for the Office Manager (username: Tara Rogers and password: password123), Technician (username: Tom Jones and password: pass123), Shift Manager (username: Chris Smith and username: 98765) and Receptionist (Username: Zoe Ball and password: ball55). | |
| **Expected results:**   1. The user logs in to the system. | |
| **Test:**   1. All the users mentioned above, enters their username and password in the respective present fields. 2. Then click the login GUI. | |
| **Test record:** The respective users logs into the system. | |
| **Date:xxx/xxx/xxx** | **Tester:xxxxxxxx** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** | |

**Alternative Flow:**

|  |  |
| --- | --- |
| **Use case ID:** 5 | **Use case name:** Login |
| **Test number:** 2 | |
| **Objective:** To test the alternative flow where the user is unable to login due to invalid/incorrect details. | |
| **Set up:**   1. An account must be made for the Office Manager (username: Tara Rogers and password: password123), Technician (username: Tom Jones and password: pass123), Shift Manager (username: Chris Smith and username: 98765) and Receptionist (Username: Zoe Ball and password: ball55). | |
| **Expected results:**   1. The user is unable to log in to the system. | |
| **Test:**   1. All the users mentioned above, enters their username and/or password incorrectly in the respective present fields. 2. They then click the login GUI. | |
| **Test record:** Login Failed prompt comes up. | |
| **Date:xxx/xxx/xxx** | **Tester:xxxxxxxx** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** | |

## Generate Reports

**Main Flow:**

|  |  |
| --- | --- |
| **Use case ID:** 12 | **Use case name:** Generate Reports |
| **Test number:** 1 | |
| **Objective:** To test the main flow | |
| **Set up:**   1. A job is setup for the customer John Doe. This job involves for the image developers, the printing team and the lamination staff. 2. Projected time frame for the job is 5 days and nights. | |
| **Expected results:**   1. Autogenerate the Individual job report. 2. Autogenerate the Individual performance report for each team/member. 3. Autogenerate the Summary Performance report for each shift (day/night). | |
| **Test:**   1. The job is active for John Doe. 2. The relevant reports are generated on time. | |
| **Test record:** Autogenerated reports are being created/provided. | |
| **Date:xxx/xxx/xx** | **Tester:xxxxxxxxxxxxxx** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** | |

## Update Job Status

**Main Flow:**

|  |  |
| --- | --- |
| **Use case ID:** 6 | **Use case name:** Update Job Status |
| **Test number:** 1 | |
| **Objective:** Test the main flow | |
| **Set up:**   1. A job is being processed for John Doe. 2. The image developers have completed their tasks. 3. The printing team are in the process of printing the images. 4. The lamination staff are awaiting go ahead signal. | |
| **Expected results:**   1. The database is updated as completed for the image developers. 2. The status for the printing team is marked as ongoing. 3. the status for the lamination staff is pending. | |
| **Test:**   1. The Office Manager/Technician/Shift Manager logs into the system. 2. They open John Doe’s job and see the corresponding job tasks, status have been updated. 3. The status of the job can be altered to fit the progress of the job. | |
| **Test record:** Update job status based on completion of task. | |
| **Date:xxx/xx/xxx** | **Tester:xxxxxxxxxxxx** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** | |

## Record Payment

**Main flow:**

|  |  |
| --- | --- |
| **Use case ID:** | **Use case name:** Record Payment |
| **Test number:** | |
| **Objective:** Test the primary path | |
| **Set up:** Customer number (\*) must be set up and have outstanding balance of (\*) of unpaid jobs. | |
| **Expected results:**   1. The System creates a payment object. 2. The System finds the outstanding (unpaid) job/s of the customer and adds them to the payment object. 3. The payment object is stored in the Database. | |
| **Test:**   1. Select the Record Payment functionality, enter the amount paid (\*), make sure that the System created and stored the payment object with correct recorded payment amount (\*). | |
| **Test record:** | |
| **Date:** | **Tester:** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

|  |  |
| --- | --- |
| **Use case ID:** | **Use case name:** Record Payment: CardPayment |
| **Test number:** | |
| **Objective:** Test the alternative path | |
| **Set up:** Customer number (\*) must be set up and have outstanding balance of (500$) of unpaid jobs. Customer card details (\*) | |
| **Expected results :**   1. The System creates a payment object. 2. The System creates card object. 3. The System finds the outstanding (unpaid) job/s of the customer and adds them to the payment object. 4. Payment object is referenced to the card object and both have correct values (\*) and (\*) accordingly. | |
| **Test:**   1. Select the Record Payment functionality, enter the amount paid (\*), Select the card payment option. Enter card number | |
| **Test record:** | |
| **Date:** | **Tester:** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

**Alternative Flow 2:**

|  |  |
| --- | --- |
| **Use case ID:** | **Use case name:** RecordPayment : NoConnectionToServer |
| **Test number:** | |
| **Objective:** Test the System’s response to lack of connection to the database server | |
| **Set up:** Customer number (\*) must be set up and have outstanding job payment of (\*). The connection to the database server must be terminated. | |
| **Expected results :**   1. The System creates a payment object. 2. The System informs the user that no communication channel can be established to the database. | |
| **Test:**   1. Select the Record Payment functionality, enter amount paid (\*), the System informs that no communication channel can be established to the database. | |
| **Test record:** | |
| **Date:** | **Tester:** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

## Generate 2nd Letter

**Main Flow:**

|  |  |
| --- | --- |
| **Use case ID:** 10 | **Use case name:** Generate 2nd letter |
| **Test number:** | |
| **Objective:** Test the main path. | |
| **Set up:** Customer number (\*) must be set up. First letter has been generated and sent to the customer. One month passes after first letter is sent and the outstanding payment (\*) is not covered. There is communication channel between terminal and printer. | |
| **Expected results:**   1. The System suspends customer account number (\*) 2. The System alerts User with user type Office Manager and generates second letter to print. 3. The System connects to a printer. 4. The System informs Office Manager the print has been completed. 5. Letter has correct values of customer number (\*) and its outstanding payment (\*) | |
| **Test:**   1. Customer account number (\*) is marked as suspended. Log in to user type Office Manager, receive alert and notification of 2nd letter generated. User confirms the print. The system connects to the printer and prints the letter with correct customer number (\*) and outstanding payment (\*) | |
| **Test record:** | |
| **Date:** | **Tester:** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

**Alternative Flow:**

|  |  |
| --- | --- |
| **Use case ID:** | **Use case name:** Create 2nd Letter: NoPrinterConnection |
| **Test number:** | |
| **Objective:** Test the System’s response to lack of printer connection. | |
| **Set up:** Customer number (\*) must be set up. First letter has been generated and sent to the customer. One month passes after first letter is sent and the outstanding payment (\*) is not covered. There is no communication channel between terminal and printer. | |
| **Expected results :**   1. The System suspends customer account number (\*) 2. The System alerts User with user type Office Manager and generates second letter to print. 3. The System informs User that there is no communication channel to the printer. | |
| **Test:**   1. Check whether customer account number (69420) is marked as suspended. Log in to user type Office Manager, receive alert and notification of 2nd letter generated. User confirms the print. The System informs user that there is no communication to the printer. | |
| **Test record:** | |
| **Date:** | **Tester:** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

## Add User

**Main Flow:**

|  |  |
| --- | --- |
| **Use case ID:** | **Use case name:** Add User |
| **Test number:** | |
| **Objective:** Test the functionality of adding a user. | |
| **Set up:** User (\*) with user type Office Manager is logged in. | |
| **Expected results:**   1. User inputs 2. Newly created user account has correct privileges. 3. Database creates a new user entry. 4. The System informs that user account has been successfully created. | |
| **Test:** 1. Select “Create User” functionality, enter details for the user account (\*), set up privileges (\*). The System informs that user has been successfully created. Check whether database record corresponds with the correct values. | |
| **Test record:** | |
| **Date:** | **Tester:** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

**Alternative Flow:**

|  |  |
| --- | --- |
| **Use case ID:** | **Use case name:** Add User: NoCommunicationChannel |
| **Test number:** | |
| **Objective:** Test the System’s response to lack of communication channel when adding a user. | |
| **Set up:** User (\*) with user type Office Manager is logged in. No communication channel with the database server. | |
| **Expected results:**   1. The System informs the user that there is no communication channel to the database server. | |
| **Test:** 1. Select “Create User” functionality, enter details for the user account (\*), set up privileges (\*). The System informs that there is no communication channel to the database server. | |
| **Test record:** | |
| **Date:** | **Tester:** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

## Automatic Backup

**Main Flow:**

|  |  |
| --- | --- |
| **Use case ID:** 9 | **Use case name:** Automatic backup |
| **Test number:** | |
| **Objective:** Test the automatic backup functionality. | |
| **Set up:** Automatic backup period (\*) is specified. | |
| **Expected results:**   1. The database server is backed up. | |
| **Test:** 1. Specified time occurs (\*), the System automatically creates new database backup. | |
| **Test record:** | |
| **Date:** | **Tester:** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

**Alternative Flow:**

|  |  |
| --- | --- |
| **Use case ID:** | **Use case name:** Automatic backup: NoCommunicationChannel |
| **Test number:** | |
| **Objective:** Test the system response to lack of communication channel when automatic backup occurs. | |
| **Set up:** Automatic backup period (\*) is specified. No communication channel to the database server. | |
| **Expected results:** The System informs that there is no communication channel to the database server. | |
| **Test:** 1. Specified time occurs (\*). The System informs the user that there is no communication channel to the database server. | |
| **Test record:** | |
| **Date:** | **Tester:** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

## Update Existing Task:

**Main Flow:**

|  |  |
| --- | --- |
| **Use case ID:** 11 | **Use case name:** Update existing task |
| **Test number:** | |
| **Objective:** Test the “Update existing task” functionality. | |
| **Set up:** Office Manager is logged in (\*). Task (\*) is created. | |
| **Expected results:**   1. Task has been updated. 2. The updated task value (\*) corresponds to the one in the database records. | |
| **Test:**   1. Select “Update existing task” functionality, select (\*) task, input new data (\*), confirm the entry. The System informs that task (\*) has been changed. Check whether new task value (\*) corresponds to the one in the database records. | |
| **Test record:** | |
| **Date:** | **Tester:** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

**Alternative Flow:**

|  |  |
| --- | --- |
| **Use case ID:** | **Use case name:** Update existing task: NoCommunicationChannel |
| **Test number:** | |
| **Objective:** Test the System’s response to lack of communication channel when updating a task. | |
| **Set up:** Office Manager is logged in (\*). No communication channel to the database server. | |
| **Expected results:**   1. The System informs user that there is no communication channel to the database server. | |
| **Test:**   1. Select “Update existing task” functionality. The System informs that there is no communication channel to the database server. | |
| **Test record:** | |
| **Date:** | **Tester:** |
| **Result:** | |
| **Date:** | **Tester:** |
| **Result:** |  |

## Non-Functional Testing

Security + Reliability to be added here

2 volere templates