# Templates

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** |
| **Brief Description:** | |
| **Primary Actors:** | |
| **Secondary Actors:** | |
| **Preconditions**: | |
| **Main Flow:** | |
| **Postconditions:** | |
| **Alternative Flows:** | |

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** |
| **Brief Description:** | |
| **Primary Actors:** | |
| **Secondary Actors:** | |
| **Preconditions**: | |
| **Alternative Flows:** | |
| **Postconditions:** | |

## Record Payment

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Record Payment |
| **Brief Description:** Once a customer makes a payment with BAPERS, a record of the payment is then stored in the local memory of the machine running the system and then the record is stored in the database. | |
| **Primary Actors:** Office Manager, Receptionist | |
| **Secondary Actors:** Database Server | |
| **Preconditions:**   1. The system is operational 2. Either the Office Manager, or the Receptionist is logged into the system and had chosen the GUI option to record a payment from a client. | |
| **Main Flow:**   1. The use case starts when the Office Manager, or receptionist selects Record Payment functionality. 2. The GUI prompts the user to then enter the payment details including: amount paid, whether the payment is card or cash. 3. The GUI then sends the input information to the system, and the system creates a payment object. 4. The system then finds the outstanding (unpaid) jobs for the customer, and adds them to the payment object. 5. The system contacts the Database Server, and sends the payment object to be stored on disk. 6. The system deletes the local payment object.   The system alerts the user that payment has been added. | |
| **Postconditions:**  The database contains a new payment record | |
| **Alternative Flows:**  CardPayment  NoConnectionToServer  UnSuspendAccount | |

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Record Payment : CardPayment |
| **Brief Description:** The customer has made a payment with a card, so the card details will be added to the system. | |
| **Primary Actors:** N/A | |
| **Secondary Actors:** N/A | |
| **Preconditions:**   1. The system is operational and Either the Office Manager, or the Receptionist is logged into the system and had chosen the GUI option to record a payment from a client. | |
| **Alternate Flow:**   1. The flow may start at step 2 of the main flow. 2. The GUI prompts the user to enter card details. 3. While (more cards)    1. The user enters card details. 4. The GUI sends the information to the system, and the system then creates a card object, and a payment object. Giving the payment object a reference to the card object.   The flow should resume at step 2 of the main flow. | |
| **Postconditions:**  Local storage contains a Card Details Object, which is referenced by a Payment object. | |

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Record Payment : NoConnectionToServer |
| **Brief Description:** A connection cannot be established with the database server, so the operation is aborted. | |
| **Primary Actors:** N/A | |
| **Secondary Actors:** N/A | |
| **Preconditions:**   1. The system is operational and either the Office Manager, or the Receptionist is logged into the system and had chosen the GUI option to record a payment from a client. | |
| **Alternate Flow:**   1. The flow may start at step 5 of the main flow. 2. The system informs the user that no communication channel can be established to the database. | |
| **Postconditions:**  The contents of the database are unchanged | |

## Generate Letter

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Generate 2nd Letter |
| **Brief Description:** BAPERS automatically generate the second letter to the Office Manager which going to be send to the customers. | |
| **Primary Actors:** Time | |
| **Secondary Actors:** Office Manager | |
| **Preconditions**:   1. First letter has been generated and sent to customer. 2. One month after the first letter is sent and payment yet to be receive. 3. Office Manager logs into his/her account. | |
| **Main Flow:**   1. The Use case starts when it is a month after the first letter sent out. 2. BAPERS suspend the corresponding customer’s account. 3. BAPERS awaits User with user type Office Manager to Log-in. 4. BAPERS alert the Office Manager and second letter is prepare for him/her to print. 5. BAPERS connects to a printer with existing communication channel. 6. BAPERS inform the Office Manager the print has been completed. | |
| **Postconditions:**  The letter successfully printed out by the Office Manager with no error occur. | |
| **Alternative Flows:**  NoPrinterConnection | |

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Generate 2nd Letter:NoPrinterConnection |
| **Brief Description:** Printing error due to lack of communication with the hardware printer. | |
| **Primary Actors:** N/A | |
| **Secondary Actors:** N/A | |
| **Preconditions**:   1. BAPERS is operational. 2. Office Manager logs into BAPERS. 3. Office Manager receive the alert of the second letter. 4. Office Manager chosen to print the second letter. | |
| **Alternative Flows:**   1. The flow may start at step 5 of the main flow. 2. BAPERS informs the Office Manager that no communication between the printer and the computer can be located. | |
| **Postconditions:**  The letter is not print on demand by the Office Manager. | |

## Create User

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Create User |
| **Brief Description:** Office Manager creates an user account for BAPERS. | |
| **Primary Actors:** Office Manager | |
| **Secondary Actors:** N/A | |
| **Preconditions**:   1. BAPERS is operational. 2. A user with user type Office Manager has logged into BAPERS. | |
| **Main Flow:**   1. The use case starts when the Office Manager selects “Create User” functionality. 2. Office Manager input details for the user account. 3. Office Manager setting up privileges for the account. 4. BAPERS connect with the database with existing communication channel and send details of the new creation account to it. 5. BAPERS informs Office Manager that a user account has been successfully created. | |
| **Postconditions:**  The database contains the new account details. | |
| **Alternative Flows:**  NoCommunicationChannel | |

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Create User:NoCommunicationChannel |
| **Brief Description:** Creation error due to lack of communication channel between BAPERS and the database. | |
| **Primary Actors:**  N/A | |
| **Secondary Actors:** N/A | |
| **Preconditions**:   1. BAPERS is operational. 2. User with user type Office Manager logged into BAPERS. 3. Office Manager chosen to create user. | |
| **Alternative Flows:**   1. The flow may start at step 4 of the main flow. 2. BAPERS informs the Office Manager that no communication channel can be located. | |
| **Postconditions:**  User account details in the database remain unchanged. | |

## Automatic Backup

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Automatic backup |
| **Brief Description:** BAPERS backs up the database server each time period specified. | |
| **Primary Actors:** Time | |
| **Secondary Actors:** Database server | |
| **Preconditions**:   1. BAPERS is operational. 2. Automatic backup period is specified. | |
| **Main Flow:**   1. Use case starts when specified time occurs. 2. BAPERS backs up the Database server. | |
| **Postconditions:**  The database server is backed up/replaces the old backup. | |
| **Alternative Flows:**  NoCommunicationChannel | |

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Automatic backup:NoCommunicationChannel |
| **Brief Description:** Automatic backup unavailable due to lack of communication channel. | |
| **Primary Actors:** N/A | |
| **Secondary Actors:** N/A | |
| **Preconditions**:   1. BAPERS is operational. 2. Automatic backup period is specified. | |
| **Alternative Flows:**   1. The flow may start at step 2 of the main flow. 2. BAPERS informs the User that no communication channel can be located | |
| **Postconditions:**  The database server is not backed up. | |

## Update Existing Task

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Update existing task |
| **Brief Description:** Edit existing tasks from the task list. | |
| **Primary Actors:** Office manager | |
| **Secondary Actors:** Database server | |
| **Preconditions**:   1. BAPERS is operational. 2. Primary actor is logged in to BAPERS. | |
| **Main Flow:**   1. Use case starts when the User selects “Update existing task” functionality. 2. Tasks are retrieved from the database server 3. User selects a task. 4. User updates the details of the task 5. BAPERS updates the task. 6. BAPERS informs user that the task has been updated. | |
| **Postconditions:**  The database server contains updated task; all previously stored tasks are unchanged. | |
| **Alternative Flows:**  NoCommunicationChannel | |

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Update existing task:NoCommunicationChannel |
| **Brief Description:** Task updating unavailable due to lack of communication channel. | |
| **Primary Actors:** N/A | |
| **Secondary Actors:** N/A | |
| **Preconditions**:   1. BAPERS is operational. 2. Primary actor is logged in to BAPERS | |
| **Alternative Flows:**   1. The flow may start at step 2 of the main flow. 2. BAPERS informs the User that no communication channel can be located | |
| **Postconditions:**  The content of tasks in database server remains unchanged. | |

## Upgrade Customer Account

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** UpgradeCustomerAccount |
| **Brief Description:** Upgrade important customers to the “Valued” status, enabling them to benefit from different discounts. | |
| **Primary Actors:** Office Manager | |
| **Secondary Actors:** Database | |
| **Preconditions**:   1. The system is functional and the Office Manager logs in to the system | |
| **Main Flow:**   1. The Use Case starts when the Office Manager logs into the system 2. The Office Manager accesses the customer list 3. The Office Manager selects the customer 4. The Office Manager clicks the GUI button for upgrading the customer to the “Valued” status 5. When the button is clicked, the system fetches the value from the database 6. The system creates an object into the system where it stores the value 7. The system changes the old value with the new value 8. The data is sent back by the system to the database 9. The database overwrites the old status with the new status. | |
| **Postconditions:**  The database is updated with the new customer status and the customer is eligible for discounts | |
| **Alternative Flows:**  InvalidAccountUpgraded | |

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** UpgradeCustomer Account:InvalidAccountUpgraded |
| **Brief Description:** The system informs the Office Manager that the account he is trying to upgrade is already upgraded | |
| **Primary Actors:** N/A | |
| **Secondary Actors:** N/A | |
| **Preconditions**:   1. The Office Manager has tried upgrading an account that is already upgraded | |
| **Alternative Flows:**   1. Alternative flow begins at step 4 2. The Office Manager clicks the GUI button for updating the user status 3. The system fetches the data from the database 4. The system compares detects that the change already took place 5. The system displays the error message 6. The system does not send any information back to the database and deletes the object created | |
| **Postconditions:**  The database is not modified | |

## Print Late Reminder

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** PrintLateReminder |
| **Brief Description:** Print the late payment reminder for the customer who did not pay in time for the service. | |
| **Primary Actors:** Office Manager | |
| **Secondary Actors:** N/A | |
| **Preconditions**:   1. The system is functional; the Office Manager logs in to the system and the customer exceeded the payment deadline | |
| **Main Flow:**   1. The screen displays the notification that a customer has exceeded the payment deadline 2. The Office Manager Clicks the GUI button for printing the letter 3. The letter is printed | |
| **Postconditions:**  The Database is updated with the information of a user having a letter printed for | |
| **Alternative Flows:**  PrinterError | |

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** PrintLateReminder: PrintLateReminder |
| **Brief Description:** Office manager is unable to print the letters required as a print error occurs | |
| **Primary Actors:** N/A | |
| **Secondary Actors:** N/A | |
| **Preconditions**:   1. The system is functional; the Office Manager logs in to the system and the customer exceeded the payment deadline | |
| **Alternative Flows:**   1. Flow may start at step 2 of the main flow 2. There is a printer error when trying to print the letter | |
| **Postconditions:**  The late reminder message is not printed | |

## Login

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Login |
| **Brief Description:** The procedure where the users/staff can login to the BAPERS system so that they can use the system and its functions. | |
| **Primary Actors:** Receptionist, Shift Manager, Office Manager, Technician | |
| **Secondary Actors:** Database server | |
| **Preconditions**:   1. System is operational, the system has a secure connection to the database server. 2. The primary actors must be authorised. | |
| **Main Flow:**   1. The use case starts when the user selects the “Login” button. 2. The system opens another page where the user is told to enter their login details in the space provided. 3. The user then presses the login button. 4. The system then searches the database for corresponding login details.    1. Corresponding login details are not found. While not found:       1. System tells user: invalid username or password.       2. System tells user to check and re-enter their username and password.    2. The user then clicks the login button again. 5. User then login 6. Use case completed. | |
| **Postconditions:**  The user login and the software and its functions are accessible to the user. | |
| **Alternative Flows:**  LoginFailed | |

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Login: LoginFailed |
| **Brief Description:** The system informs the user that login failed due to an incorrect username or password or both. | |
| **Primary Actors:** N/A | |
| **Secondary Actors:** N/A | |
| **Preconditions**:   1. The user entered an incorrect username and/or password, which was not found in the database. | |
| **Alternative Flows:**   1. The alternative flow begins at step 4 of the main step. 2. The system informs the user that invalid details have been provided. | |
| **Postconditions:**  The user is denied access to the system | |

Generate Reports

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Generate Report |
| **Brief Description:** Various reports are generated automatically by BAPERS | |
| **Primary Actors:** Time | |
| **Secondary Actors:** Database server | |
| **Preconditions**:   1. BAPERS is operational 2. There is an established connection to the database server. | |
| **Main Flow:**   1. The use case starts when the time triggers the functionality of generating reports. 2. Individual report for the jobs brought in by a particular customer for an arbitrary period is taken. 3. Individual performance report on work undertaken by a member of BIPL staff is generated 4. Summary Performance report for work undertaken by BIPL during day and night shifts is generated. | |
| **Postconditions:**  The generated reports get stored in the database | |
| **Alternative Flows:**  None | |

Update Job Status

|  |  |
| --- | --- |
| **Use Case ID:** | **Use Case:** Update Job Status |
| **Brief Description:** primary actor updates the status of an ongoing job | |
| **Primary Actors:** Office Manager, Shift Manager, Technician | |
| **Secondary Actors:** N/A | |
| **Preconditions**:   1. The system is operational. 2. The primary actor is already logged in. | |
| **Main Flow:**   1. The use case starts when the Manager or Technician chooses to update the status of the job 2. They can record the completion of the current task which is shown as an include 3. They can also update the system with the commencement of the next task 4. Once updated, the status is recorded and sent back to the database. | |
| **Postconditions:**  Completion of tasks are recorded | |
| **Alternative Flows:** | |