12/12/2014

Software Engineering: Catherine Woods

**CompSys**

Nazmul Alam (TL\_KCPSD\_DY2)

T00152975

Table of Contents

1. Introduction 3

2. Functional Components 4

2.1 Hierarchy Chart 4

3. User Requirements 5

3.1 The system performs staff administration 5

3.1.1 CompSys will allow the details of each new staff to be recorded. 5

3.1.2 CompSys will allow a staff details to be amended. 5

3.1.3 CompSys will generate a staff listing. 5

3.2 The system performs customer administration 5

3.2.1 CompSys will allow the details of each new customer to be recorded. 5

3.2.2 CompSys will allow a customer details to be amended. 5

3.2.3 CompSys will allow a customer to be queried. 5

3.2.4 CompSys will generate a customer listing. 5

3.3 The system performs repair management 5

3.3.1 CompSys will allow the details of each new repair to be recorded. 5

3.3.2 CompSys will generate a new repair estimate for a new repair. 5

3.3.3 CompSys will allow the system to validate a repair. 5

3.3.4 CompSys will allow the system to finalise a repair. 5

3.3.5 CompSys will process every repair being collected. 5

3.3.6 CompSys will generate repair report for all the completed repairs. 5

3.4 The system performs administration 5

3.4.1 CompSys will generate a list of jobs and their status. 5

3.4.2 CompSys will issue invoice for any repair at any given period. 5

3.4.3 CompSys will generate income analysis to required specification. 5

3.4.4 CompSys will allow the system to set pre-defined repair rates and parts in the inventory. 5

4. System Requirements 6

4.1. System Level Use Case Diagram 6

4.2. Manage Staff 7

4.2.1 New Staff 7

4.2.2 Amend Staff 11

4.2.3 List Staff 15

4.3. Manage Customers 17

4.3.1 New Customer 17

4.3.2 Amend Customer 20

4.3.3 Query Customer 24

4.3.4 List Customers 27

4.4. Manage Repair 29

4.4.1 New Repair 29

4.4.2 Set Estimate 33

4.4.3 Validate Repair 37

**4.4.4** **Finalise Repair** 39

4.4.5 Collect Repair 41

4.4.6 Repair Report 44

4.5. Manage Admin 46

4.5.1 List Repair Jobs 46

4.5.2 Issue Invoice 49

4.5.3 Income Analysis 51

4.5.4 Inventory 53

5. System Model 56

5.1. Data Flow Diagrams 56

5.1.1 Level 0 DFD 57

5.1.2 Level 1 DFD 58

5.1.3 Level 2 DFD (Process P1 – Manage Staff) 59

5.1.4 Level 2 DFD (Process P2 – Manage Customers) 60

5.1.5 Level 2 DFD (Process P3 – Manage Repairs) 61

5.1.6 Level 2 DFD (Process P4 – Manage Admin) 62

6. Data Model 63

6.1. UML Class Diagram 63

7. Database Schema 64

7.1 Relational Schema 64

7.2 Schema: CompSys 65

7.2.1 Relation: Customers 65

7.2.2 Relation: Staff 65

7.2.3 Relation: Repairs 66

7.2.4 Relation: Estimates 66

7.2.5 Relation: Parts 66

7.2.6 Relation: EstParts 67

8. Conclusion 68

# **Introduction**

CompSys is s computer repair system software to maintain everyday task for a computer repair shop. The software provides cutting-edge technology to make your daily task a lot more easier.

CompSys aims to be a simple, fast and technician friendly way of managing your computer repair store. Our goal is to be powerful in functionality while simplistic in appearance, no clutter or other bloated features.

**Features**

* Tracking of Repair Workflow
* Customer Info & Assets
* Support for Multiple Tax rates on both Goods and Services and multiple taxes per item.
* System Based Interface
* Printable Claim Ticket, Invoices, Checkout Receipt, Sale Receipts & Repair Report
* Email Invoices, Repair Reports, Receipts
* Receipt Printer Support
* Reporting - Sales and Technician Activity
* Oracle SQL Database driven
* Store Inventory Management
* Multi-Store Support
* Job Timers
* Block of Time Contracts
* Recurring Invoices
* Email Service Reminders to your customers
* Translatable - You can translate the interface from English to your language right in the interface.

# **Functional Components**

## **Hierarchy Chart**

# **User Requirements**

## **The system performs staff administration**

### CompSys will allow the details of each new staff to be recorded.

### CompSys will allow a staff details to be amended.

### CompSys will generate a staff listing.

## **The system performs customer administration**

### CompSys will allow the details of each new customer to be recorded.

### CompSys will allow a customer details to be amended.

### CompSys will allow a customer to be queried.

### CompSys will generate a customer listing.

## **The system performs repair management**

### CompSys will allow the details of each new repair to be recorded.

### CompSys will generate a new repair estimate for a new repair.

### CompSys will allow the system to validate a repair.

### CompSys will allow the system to finalise a repair.

### CompSys will process every repair being collected.

### CompSys will generate repair report for all the completed repairs.

## **The system performs administration**

### CompSys will generate a list of jobs and their status.

### CompSys will issue invoice for any repair at any given period.

### CompSys will generate income analysis to required specification.

### CompSys will allow the system to set pre-defined repair rates and parts in the inventory.

# **System Requirements**

## **System Level Use Case Diagram**

**Computer Repair System**

Manager

Staff

## **Manage Staff**

This module is responsible for adding new staff, amending existing staff and listing them in ascending order of surname, forename. The staff can be set to **inactive** from the ‘amend staff’ sub-module.

### New Staff

#### UC Diagram to allow the details of each new staff to be recorded.

Manager

Staff

<<includes>>

<<extends>>

<<includes>>

<<requires>>

#### UC Narrative to allow the details of each new staff to be recorded.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **New Staff** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.1.1 | | **Date: 14/10/2014** |
| **Priority** | High | | |
| **Source** | Manager | | |
| **Primary Business Actor** | Manager | | |
| **Other Participating Actors** | Staff | | |
| **Description** | * This function registers new staff in the system. | | |
| **Preconditions** |  | | |
| **Trigger** | * Manager needs to register a new staff member | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager chooses add new staff.  **Step 3:** The manager gets staff details:   * **Firstname** * **Surname** * **DOB** * **Staff Type** * **Mobile/Tel Number** * **Email Address** * **Address 1** * **Address 2** * **Town** * **County** * **Username** * **Password**   **Step 4:** The manager confirms that the staff is to be registered | **Step 2:** The system displays create staff form.  **Step 5:** The system validates the data entered:   * All the fields but **Address 2** are required. * **Firstname, Surname:**   + Should should only be letters   + One space between every words   + No trailing spaces   + Between 1-30 characters * **Mobile number** should only be digits, no spaces and between 7-10 characters * **Email** should not contain any special characters or any spaces.   + Only hyphen, underscore and comma is allowed.   + Alphanumeric characters only.   + Should not be greater than 100 chracters long * **Address 1, Address 2** should be alphanumeric characters only.   + No leading or trailing spaces   + Should not be greater than 50 characters long   + Only one space beteween every words * **Town** should be contain only letters   + No leading or trailing spaces   + One space between every words   + Between 1-30 characters * **Username** should be 6-10 characters long and can only be alphanumeric * **Password** should be between 5-20 characters long   + Contain at least 2 letters, 2 digits, 1 uppercase, 1 lowercase character * **Staff age** must be between 16 - 60 years   **Step 6:** The system assigns the **Staff ID**  **Step 7:** The system assigns **Registration Date** as the current system date  **Step 8:** The system assigns the staff status a default value of **‘A’**  **Step 9:** The system saves the new staff details in the Staff file.  **Step 10:** The system displays a confirmation message and asks if the manager is interested in adding a new staff. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Invalid and/or Required field NOT entered** | **Step 7:** Manager re-enters the required field. | **Step 5:** A required field is not entered  **Step 6:** The system displays an appropriate error message and notifies the error sources | |
| **A staff member is created** | | |
| **Conclusions** | * New staff is created. | | |
| **Post conditions** | * New staff can now do repair work. | | |
| **Business Rules** |  | | |
| **Implementation Constraints** |  | | |

### Amend Staff

#### UC Diagram to allow staff details to be amended.

Staff

Manager

<<extends>>

<<includes>>

<<includes>>

#### UC Narrative to allow staff details to be amended.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Amend Staff** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.1.2 | | **Date: 22/10/2014** |
| **Priority** | Medium | | |
| **Source** | Staff | | |
| **Primary Business Actor** | Manager | | |
| **Other Participating Actors** | Staff | | |
| **Description** | * This function updates staff details and save them to the staff file. | | |
| **Preconditions** | * New staff should supply updated details. | | |
| **Trigger** | * Staff requests staff details to be amended. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** Staff makes an application to amend details of the staff file and submits new staff details.  **Step 3:** Manager requires these details to query for the staff:   * **Firstname** * **Lastname** * **Town**   **Step 6:** The manager chooses the appropriate staff from the list.  **Step 9:** The manager amends the new staff details   * **Firstname** * **Surname** * **DOB** * **Staff Type** * **Mobile/Tel Number** * **Email Address** * **Address 1** * **Address 2** * **Town** * **County** * **Username** * **Password** | **Step 2:** Sytem generates autocomplete lists from the database and displays a query form.  **Step 4:** System validates data entry.   * Firstname and Lastname are mandatory fields. * **Firstname, Lastname:**   + Should should only be letters   + One space between every words   + No trailing spaces   + Between 1-30 characters * **Town** should be contain only letters   + No leading or trailing spaces   + One space between every words   + Between 1-30 characters   **Step 5:** The system retrieves staff details based on the information queried from the staff file and displays a list of matching staff.  **Step 7:** System confirms if the staff/manger wants to amend the chosen staff.  **Step 8:** The system displays amend staff form pre-fillled with the choosen staff information retrieved from the staff file.  **Step 10:** The system validates the data entered:   * All the fields but **Address 2** are required. * **Firstname, Surname:**   + Should should only be letters   + One space between every words   + No trailing spaces   + Between 1-30 characters * **Mobile number** should only be digits, no spaces and between 7-10 characters * **Email** should not contain any special characters or any spaces.   + Only hyphen, underscore and comma is allowed.   + Alphanumeric characters only.   + Should not be greater than 100 chracters long * **Address 1, Address 2** should be alphanumeric characters only.   + No leading or trailing spaces   + Should not be greater than 50 characters long   + Only one space beteween every words * **Town** should be contain only letters   + No leading or trailing spaces   + One space between every words   + Between 1-30 characters * **Username** should be 6-10 characters long and can only be alphanumeric * **Password** should be between 5-20 characters long   + Contain at least 2 letters, 2 digits, 1 uppercase, 1 lowercase character * **Staff age** must be between 16 - 60 years   **Step 11:** The system amends the new staff details in the Staff file**.**  **Step 12:** A Message is displayed to the screen to inform the manager that the staff details have been updated and asks if the manager is interested in amending another staff. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Invalid and/or Required field NOT entered** | **Step 6:** Manager re-enters the required field. | **Step 4 & 10:** Invalid data and/or required field is not entered  **Step 5:** The system displays an appropriate error message and notifies the error sources | |
| **Staff details are amended.** | | |
| **Conclusions** | * The staff details are amended. | | |
| **Post conditions** |  | | |
| **Business Rules** |  | | |
| **Implementation Constraints** |  | | |

### List Staff

#### UC Diagram to generate a staff listing.

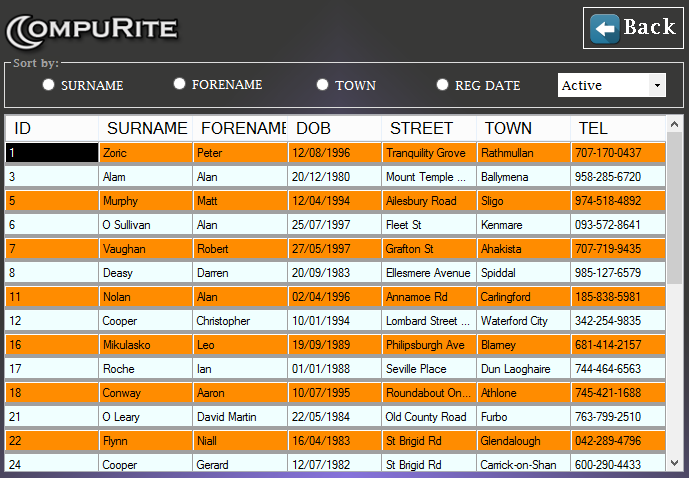
Staff

<<includes>>

<<includes>>

<<includes>>

#### Sample listing data



#### UC Narrative to generate a staff listing.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **List Staff** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.1.3 | | **Date: 24/10/2014** |
| **Priority** | Low | | |
| **Source** | Staff | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** |  | | |
| **Description** | This function generates a staff listing. | | |
| **Preconditions** | * Staff details should exist in the system. | | |
| **Trigger** | * Shop manager wants to list all the staff. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager/staff requests a list of all the staff.  **Step 3:** The manager/staff chooses display by options:   * **Surname** * **Forname** * **Town** * **Registration Date** * **Active/Inactive** | **Step 2:** The system retrieves details of all current staff from the **staff file** and displays them.  **Step 4:** The system generates a list according to the option chosen by the manager/staff and displays on the system. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Manager/Staff wants to print** | **Step 4:** The manager requests a printed copy of the staff list. | **Step 5:** System confirms if the staff/mangaer wants to print the listing or preview the listing before printing.  **Step 6:** The system generates a printed list based on the retrieved data and sends to the default print device. | |
| **Conclusions** | * A listing of all the staff is generated/printed. | | |
| **Post conditions** | * Details of the staff displayed cannot be amended or deleted. | | |
| **Business Rules** | * Staff details may only be viewed, not changed or deleted, when displayed in this way. * The list should include only current staff. | | |

## **Manage Customers**

This module is responsible for adding new customer, querying and amending existing customer accounts and listing them in ascending order of surname, forename. The customer can be set to **inactive** from the ‘amend customer’ sub-module.

### New Customer

#### UC Diagram to allow the details of each new customer to be recorded.

Staff

Customer

<<includes>>

<<extends>>

<<includes>>

<<includes>>

#### UC Narrative to allow the details of each new customer to be recorded.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **New Customer** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.2.1 | | **Date: 24/10/2014** |
| **Priority** | High | | |
| **Source** | Manager, Staff | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** | Customer | | |
| **Description** | * This function registers a new customer in the system. | | |
| **Preconditions** |  | | |
| **Trigger** | * Shop staff/manager needs to register a new customer. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The staff/manager chooses add new customer.  **Step 3:** The staff/manager enters the required information:   * **Firstname** * **Surname** * **DOB** * **Customer Type** * **Mobile/Tel Number** * **Email Address** * **Address 1** * **Address 2** * **Town** * **County**   **Step 4:** The staff/manager confirms that the staff is to be registered | **Step 2:** The system displays create customer form.  **Step 5:** The system validates the data entered:   * All the fields but **Address 2** are required. * **Firstname, Surname:**   + Should should only be letters   + One space between every words   + No trailing spaces   + Between 1-30 characters * **Mobile number** should only be digits, no spaces and between 7-10 characters * **Email** should not contain any special characters or any spaces.   + Only hyphen, underscore and comma is allowed.   + Alphanumeric characters only.   + Should not be greater than 100 chracters long * **Address 1, Address 2** should be alphanumeric characters only.   + No leading or trailing spaces   + Should not be greater than 50 characters long   + Only one space beteween every words * **Town** should be contain only letters   + No leading or trailing spaces   + One space between every words   + Between 1-30 characters * **Customer age** must be between 16 - 60 years   **Step 6:** The system assigns the **Customer ID**  **Step 7:** The system assigns **Registration Date** as the current system date  **Step 8:** The system assigns the customer status a default value of **‘A’**  **Step 9:** The system saves the new customer details in the **Customer File.**  **Step 10:** The system displays a confirmation message and asks the staff/manager wants to add another customer. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Invalid data type and/or Required field NOT entered** | **Step 5c:** Staff/Manager re-enters the required field. | **Step 5:** A required field is not entered and/or invalid data type.  **Step 6:** The system displays an appropriate error message and notifies the error sources | |
| **A new customer is created** | | |
| **Conclusions** | * New customer is created in the customer file. | | |
| **Post conditions** | * New customer can now proceed with repair work | | |

### Amend Customer

#### UC Diagram to allow customer details to be amended.

Staff

Customer

<<extends>>

<<includes>>

<<includes>>

<<includes>>

#### UC Narrative to allow customer details to be amended.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Amend Customer** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.2.2 | | **Date: 22/10/2014** |
| **Priority** | Medium | | |
| **Source** | Staff, Manager | | |
| **Primary Business Actor** | Manager | | |
| **Other Participating Actors** | Staff | | |
| **Description** | * This function updates customer details and save them to the customer file. | | |
| **Preconditions** | * New customer should supply updated details. | | |
| **Trigger** | * Customer requests ‘customer details’ to be amended. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** Customer makes an application to amend details of the customer file and submits new customer details.  **Step 3:** Manager/staff requires these details to query for the customer:   * **Surname** * **Forename** * **Town**   **Step 6:** Manager chooses the appropriate customer to amend.  **Step 9:** The manager amends the required details:   * **Firstname** * **Surname** * **DOB** * **Customer Type** * **Mobile/Tel Number** * **Email Address** * **Address 1** * **Address 2** * **Town** * **County** | **Step 2:** Sytem generates autocomplete lists from the database and displays a query form.  **Step 4:** System validates data entry.   * Firstname and Lastname are mandatory fields. * **Firstname, Lastname:**   + Should should only be letters   + One space between every words   + No trailing spaces   + Between 1-30 characters * **Town** should be contain only letters   + No leading or trailing spaces   + One space between every words   + Between 1-30 characters   **Step 5:** The system retrieves customer details based on the information queried from the customer file and displays a list of matching customers.  **Step 7:** System confirms if the staff/manger wants to amend the chosen customer  **Step 8:** The system displays amend customer form pre-fillled with the choosen customer information retrieved from the customer file.  **Step 10:** The system validates the data entered:   * All the fields but **Address 2** are required. * **Firstname, Surname:**   + Should should only be letters   + One space between every words   + No trailing spaces   + Between 1-30 characters * **Mobile number** should only be digits, no spaces and between 7-10 characters * **Email** should not contain any special characters or any spaces.   + Only hyphen, underscore and comma is allowed.   + Alphanumeric characters only.   + Should not be greater than 100 chracters long * **Address 1, Address 2** should be alphanumeric characters only.   + No leading or trailing spaces   + Should not be greater than 50 characters long   + Only one space beteween every words * **Town** should be contain only letters   + No leading or trailing spaces   + One space between every words   + Between 1-30 characters * **Customer age** must be between 16 - 60 years   **Step 11:** The system amends the new customer details in the Customer File.  **Step 12:** A Message is displayed to the screen to inform the details have been updated and asks if the manager/staff wants to amend another customer. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Invalid and/or Required field NOT entered** | **Step 6:** Manager/Staff re-enters the required field. | **Step 4 & 10:** Invalid data and/or required field is not entered  **Step 5:** The system displays an appropriate error message and notifies the error sources | |
| **Customer details are amended** | | |
| **Conclusions** | * The customer details are amended. | | |
| **Post conditions** |  | | |
| **Business Rules** |  | | |
| **Implementation Constraints** |  | | |

### Query Customer

#### UC Diagram to allow querying a customer.

Customer

Manager

Staff

<<requires>>

<<includes>>

<<includes>>

#### UC Narrative to allow querying a customer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case Name** | **Query Customer** | | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.2.3 | | | **Date: 22/10/2014** |
| **Priority** | Medium | | | |
| **Source** | Staff, Manager | | | |
| **Primary Business Actor** | Staff | | | |
| **Other Participating Actors** |  | | | |
| **Description** | * The customer requests a detailed record of all account transactions over a specific period | | | |
| **Preconditions** |  | | | |
| **Trigger** | * Customer requires detailed customer information. | | | |
| **Typical Scenario** | **Actor Action** | **System Response** | | |
|  | **Step 1:** Staff wants a detailed description of all account transactions of a specific customer  **Step 3:** Staff requires these information:   * **Surname** * **Forename** * **Town**   **Step 6:** Manager/Staff chooses the appropriate customer.  **Step 9:** Staff/Manager reviews Customer Profile and prints a detailed description of transactions over the specified time period | **Step 2:** System generates autocomplete lists from the database and displays the query form on the UI.  **Step 4:** System validates data entry.   * Firstname and Lastname are mandatory fields. * **Firstname, Lastname:**   + Should should only be letters   + One space between every words   + No trailing spaces   + Between 1-30 characters * **Town** should be contain only letters   + No leading or trailing spaces   + One space between every words   + Between 1-30 characters   **Step 5:** The system retrieves customer details based on the information queried from the customer file and displays a list of matching customers.  **Step 7:** System confirms if the staff/manger wants to query the chosen customer  **Step 8:** System retrieves details from the **Customers file, Repairs file and Estimate file**  **Step 9:** System displays customers details, total invoices, invoiced balance, overdue balance, repairs listing, estimate listing and displays on the UI.  **Step 10**: A detailed account record is printed for the Customer | | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | | |
| **Invalid and/or Required field NOT entered** | **Step 6:** Manager/Staff re-enters the required field. | | **Step 4:** Invalid data and/or required field is not entered  **Step 5:** The system displays an appropriate error message and notifies the error sources | |
|  | | | |
| **Conclusions** | * Manager/staff retrieves customer details. * Customer details are printed. | | | |
| **Post conditions** | * Customer requires account summary | | | |
| **Business Rules** |  | | | |
| **Implementation Constraints** |  | | | |

### List Customers

#### UC Diagram to generate a customer listing.

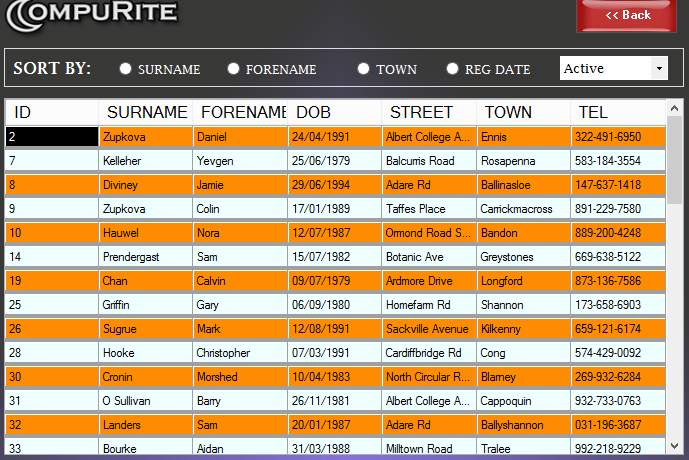
Staff

<<includes>>

<<includes>>

<<includes>>

#### Sample Listing Data



#### UC Narrative to generate a customer listing.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **List Customers** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.3.4 | | **Date: 24/10/2014** |
| **Priority** | Low | | |
| **Source** | Staff | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** |  | | |
| **Description** | This function generates a customer listing. | | |
| **Preconditions** | * Customer details should exist in the system. | | |
| **Trigger** | * Shop manager/staff wants to list all the customer | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager/staff requests a list of all the customers.  **Step 3:** The manager/staff chooses display in order of:   * **Surname** * **Forname** * **Town** * **Registration Date** * **Active/Inactive** | **Step 2:** The system retrieves details of all customer from the **customer file** and displays them on the UI.  **Step 4:** The system generates a list according to the option chosen by the manager/staff and displays on the system. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Manager/Staff wants to print** | **Step 4:** The manager requests a printed copy of the customers list. | **Step 5:** System confirms if the staff/mangaer wants to print the listing or preview the listing before printing  **Step 6:** The system generates a printed list based on the retrieved data and sends to the default print device. | |
| **Conclusions** | * A listing of all the customers is generated/printed. | | |
| **Post conditions** | * Details of the customers displayed cannot be amended or deleted. | | |
| **Business Rules** | * Customer details may only be viewed, not changed or deleted, when displayed in this way. * The list should include only current customers. | | |

## **Manage Repair**

This module is responsible for all the repairs. CompSys will allow the staff/manager to enter **New Repair** and then generate a **New Estimate** of the repair. Once the estimate is generated, the system will print out the estimate or email it to the customer. If the customer is happy with the estimate and validates the repair process to go through, the system will allow the staff/manager to change the repair status to ‘**accepted**’ (or ‘**denied**’ if otherwise) in the **Validate Repair** sub-module.

The system will allow the staff/manager to update the repair progress status (i.e. **fixed, not-fixable, in-progress, parts-required**) in the Validate Repair sub-module. If the repair is **fixed or parts-required** the CompSys system will notify the customer.

The repair collection status can be updated from the **Collect Repair** sub-module.

Once the repair is ‘**collected**’ the system will generate a new invoice from the Collect Repair sub-module and repair reports from the **Repair Reports** sub-module.

### New Repair

#### UC Diagram to allow the details of each new repairs to be recorded.

Staff

Customer

<<includes>>

<<extends>>

<<includes>>

<<includes>>

<<includes>>

#### UC Narrative to allow the details of each new repairs to be recorded.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **New Repair** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.3.1 | | **Date: 24/10/2014** |
| **Priority** | High | | |
| **Source** | Manager, Staff | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** | Customer | | |
| **Description** | * This function records details of a repair requested by a customer. | | |
| **Preconditions** | * Customer should be registered in the system. | | |
| **Trigger** | * Shop staff/manager needs to register a new repair. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** Customer brings a new repair.  **Step 3:** Manager/staff requires these details to query for the customer:   * **Surname** * **Forename** * **Town**   **Step 6:** Manager/staff chooses the appropriate customer.  **Step 9**: The staff/manager enters repair details:   * **Device Type** * **Brand** * **Model** * **OS** * **Problem Type** * **Description** | **Step 2:** Sytem generates autocomplete lists from the database and displays a query form.  **Step 4:** System validates data entry.   * Firstname and Lastname are mandatory fields. * **Firstname, Lastname:**   + Should should only be letters   + One space between every words   + No trailing spaces   + Between 1-30 characters * **Town** should be contain only letters   + No leading or trailing spaces   + One space between every words   + Between 1-30 characters   **Step 5:** The system retrieves customer details based on the information queried from the customer file and displays a list of matching customers.  **Step 7:** System confirms if the staff/manger wants to add a new repair for the chosen customer  **Step 8:** System displays repair form on the UI.  **Step 10:** System validates the data entered:   * All fieleds except **Problem Type** are required. * **Brand, Model:**   + Should should only be letters/numbers   + One space between every words   + No trailing spaces   + Between 1-50 characters * **Description** can only be letters/numbers, between 1-800 characters long, only one space between words and no trailing spaces.   **Step 11:** The system assigns the **Repair ID**  **Step 12:** The system assigns **Repair Date** as the current system date  **Step 13:** The system assigns the repair status a default value of ‘**New**’.  **Step 14:** The system saves the new repair details in the **Repairs File*.***  **Step 15:** The system displays a confirmation message | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Invalid Data** | **Step 6:** Staff/Manager re-enters the required field. | **Step 4 & 10:** A required field is not entered.  **Step 5:** The system displays an appropriate error message and notifies the error sources | |
| **New repair is created.** | | |
| **Conclusions** | * New repair is created in the repairs file. | | |
| **Post conditions** | * New repair created on the system cannot be deleted. | | |
| **Business Rules** | * Staff/Manager cannot produce a new estimate without a new repair. | | |
| **Implementation Constraints** |  | | |

### Set Estimate

#### UC Diagram to generate a new repair estimate.

<<includes>>

Staff

Customer

<<includes>>

<<extends>>

<<includes>>

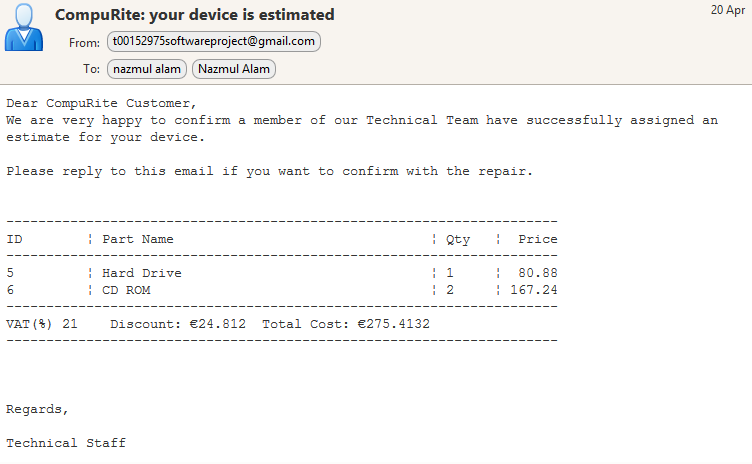
<<includes>>

<<includes>>

#### UC Narrative to generate a new repair estimate.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case Name** | **Set Estimate** | | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.3.2 | | | **Date: 24/10/2014** |
| **Priority** | High | | | |
| **Source** | Manager, Staff | | | |
| **Primary Business Actor** | Staff | | | |
| **Other Participating Actors** | Customer | | | |
| **Description** | * This function generates estimates for a new repair. | | | |
| **Preconditions** | * Customer should be registered in the system. | | | |
| **Trigger** | * Shop staff/manager needs to get estimate for a new repair. | | | |
| **Typical Scenario** | **Actor Action** | | **System Response** | |
|  | **Step 1:** Staff/manager invokes the assign estimate function.  **Step 3:** Staff/manager selects required repair from list displayed.  **Step 6**: The manager/staff enters **Part Name** and **Description**  **Step 9:** Manager/staff enters quantity needed and adds it to the shopping cart.  **Step 13:** Staff/manager gets the total estimate and confirms the estimate to be assigned. | | **Step 2:** System retrieves summary details of all ‘**New**’ repairs from **Repairs file** and associated customer details from **Customers File** and displays on UI  **Step 4:** System retrieves all details of selected repair and associated customer then displays estimate form on the UI  **Step 5:** System generates autocomplete lists data from the **Parts file.**  **Step 7:** System validates the entry.   * Both fields are required.   **Step 8:** System retrieves the part detail and displays on the UI.  **Step 10:** System validates the entry.   * Must be a number and a mandatory field. * Quantity needed should be less than or equal to the quantity available. * Part\_ID should be unique i.e. only one part matching with the Part\_ID could be added to the cart   **Step 11:** System generates sub-total on each part added/updated and displays on the UI.  **Step 12:** System adjusts the quantity and price on each item added/updated.  **Step 14:** System validates   * The total cost is not empty   **Step 15:** System generates a new **Estimate Id**.  **Step 16:** The system assigns the estimate status to ‘**Estimated’** in the **Repairs file.**  **Step 17:** System sets system date as the EstimatedDate.  **Step 18:** The system saves the estimate details in the **Estimate File.**  **Step 19:** System retrieves the email address associated with the customer.  **Step 20:** System generates all the data to email friendly version and sends it to the customer.  **Step 21:** System updates the quantity (QuantityAvailable – Quantity Used in the Estimate) to the **Parts file.**  **Step 22:** The system displays a confirmation message. | |
| **Alternate Scenarios** | **Actor Action** | | **System Response** | |
| **Delete cart item**  **Quantity entered is not valid** | **Step 12**: The manager/staff wants to delete item from the cart. | | **Step 13:** System deletes the items selected and updates the sub-total | |
| **Step 12:** Manager/staff enteres quantity needed again. | | **Step 10:** Quantity details entered is not vaild.  **Step 11:** System displays error message to the UI. | |
| **Total is empty** | **Step 16:** Staff/manager adds the total cost or adds items to the cart. | | **Step 14:** Total cost is empty.  **Step 15:** System displays an error mssages on the UI. | |
|  | | | |
| **No part is found** | **Step 6:** No applicable part is found. | **Step 7:** System displays an error message. | | |
|  | | | |
| **Conclusions** | * Estimate is recorded for a repair. | | | |
| **Post conditions** |  | | | |
| **Business Rules** | * Estimate can only be issued for New Repairs. | | | |

#### Sample Email output.



### Validate Repair

#### UC Diagram to allow the system to validate a new repair.

Staff

Customer

<<includes>>

<<includes>>

<<includes>>

#### UC Narrative to allow the system to validate a new repair.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Validate Repair** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.3.3 | | **Date: 24/10/2014** |
| **Priority** | High | | |
| **Source** | Manager, Staff | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** | Customer | | |
| **Description** | * This function validates a new repair. | | |
| **Preconditions** | * Estimate file should exit * Shop manager/staff should contact the customer for approval. | | |
| **Trigger** | * Shop staff/manager needs to validate a new estimate | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The staff/manager wants to validate an estimate.  **Step 3:** The staff/manager selects required repair from the list displayed and validates. | **Step 2:** System retrieves summary details of active repairs from **Repairs file & Estimate file** and associated customer details from **Customers File** and displays on UI  **Step 4:** System confirms if the selected repairs should be validated.  **Step 5:** System updates the status of the repair to ‘Validated’ to the Repairs file.  **Step 6**: System displays a confirmation message. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Print repair listing** | **Step 3:** Manager/staff wants to print repair listing. | **Step 4:** System generates printer friendly listing and confirms if the listing to be previewed before printing.  **Step 5:** System sends the print file to the printer device. | |
| **Conclusions** | * Estimate is validated | | |
| **Post conditions** |  | | |
| **Business Rules** | * Repairs cannot proceed without approval from customer. | | |

* + 1. **Finalise Repair** 
       1. ***UC Diagram to allow the system to finalise repair.***

Staff

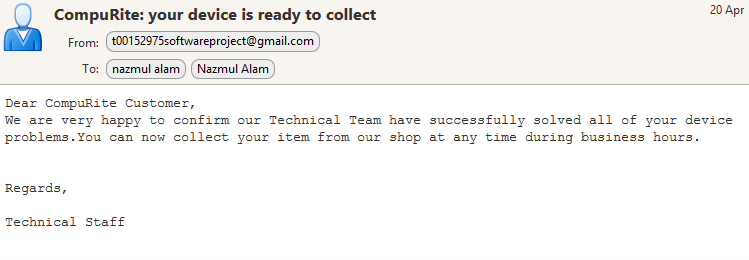
Customer

<<includes>>

<<includes>>

<<includes>>

#### Sample Email Output



* + - 1. ***UC Narrative to allow the system to finalise a new repair.***

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Finalise Repair** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.3.4 | | **Date: 24/10/2014** |
| **Priority** | High | | |
| **Source** | Manager, Staff | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** | Customer | | |
| **Description** | * This function finalises a new repair. | | |
| **Preconditions** |  | | |
| **Trigger** | * Shop staff/manager needs to finalise a repair. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The staff/manager wants to finalise a repair.  **Step 3:** The staff/manager selects required repair from the list displayed and finalises. | **Step 2:** System retrieves summary details of active repairs from **Repairs file, Estimates file** and associated customer details from **Customers File** and displays on UI  **Step 4:** System confirms if the selected repair should be finalised.  **Step 5:** System updates the repair status to ‘Invoiced’ to the Repairs file.  **Step 6:** System retrieves the email address assigned to the customers associated the finalised repairs.  **Step 7:** System generates all the data to email friendly version and sends it to the customer.  **Step 8**: System displays a confirmation message. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Print repair listing** | **Step 3:** Staff/manager wants to print repair listing. | **Step 4:** System generates printer friendly listing and confirms if the listing to be previewed before printing.  **Step 5:** System sends the print file to the printer device. | |
| **Conclusions** | * Repair is finalised. | | |
| **Post conditions** |  | | |

### Collect Repair

#### UC Diagram will process every repair being collected.

Staff

Customer

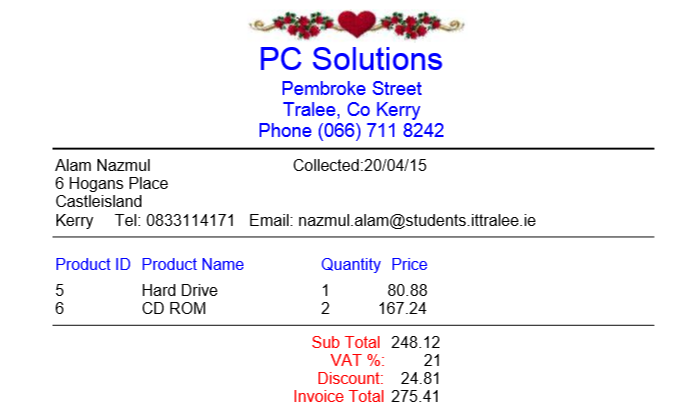
<<includes>>

<<includes>>

#### UC Narrative will process every repair being collected

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Collect Repair** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.3.5 | | **Date: 24/10/2014** |
| **Priority** | High | | |
| **Source** | Manager, Staff | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** | Customer | | |
| **Description** | * This function deals with payments and invoice. | | |
| **Preconditions** | * Repair file should exist * Payment should be provided | | |
| **Trigger** |  | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** A customer wants to collect a repair.  **Step 3:** The staff/manager selects required repair from the list displayed.  **Step 5:** Customer provides payment.  **Step 8:** A manager/staff wants to print a receipt. | **Step 2:** System retrieves summary details of all ‘**Invoiced**’ repairs from **Repairs file** & **Estimate file** and associated customer details from **Customers File** and displays on UI  **Step 4:** System retrieves all details of selected repair/customer and displays on UI  **Step 6:** A record of the transaction will be entered into the **Repairs File**. The customer’s account file is updated.  **Step 7:** A confirmation message is displayed.  **Step 9:** System generates a printer-friendly receipt and sends it to the default printer.  **Step 10:** System changes the status of the repair to ‘**Paid**’.  **Step 11**: Systems assigns **Collection Date** to current system date in the Repairs file.  **Step 12:** System updates the Repairs file. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
|  |  |  | |
|  | | |
| **Conclusions** | * Customer collects a repair. * Staff/manager finalises a repair and issues an invoice. | | |
| **Post conditions** |  | | |
| **Business Rules** | * All transactions can’t be complete until all customer details have been obtained. | | |
| **Implementation Constraints** |  | | |

#### Sample receipt



### Repair Report

#### UC Diagram to repair report for each repair.

<<includes>>

Staff

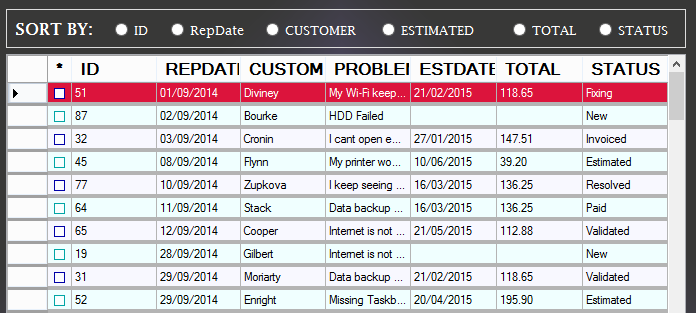
Customer

<<requires>>

<<includes>>

<<includes>>

#### Sample data of the produced report.



#### UC Narrative to generate repair report for each repair.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Repair Reports** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.3.6 | | **Date: 24/10/2014** |
| **Priority** | High | | |
| **Source** | Manager, Staff | | |
| **Primary Business Actor** | Staff | | |
| **Other Participating Actors** | Manager | | |
| **Description** | * A manager/staff generates a repair report. | | |
| **Preconditions** |  | | |
| **Trigger** |  | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1**: Shop manager/staff wants to generate a repair report.  **Step 4:** Shop manager/staff wants to print out a repair report. | **Step 2:** System retrieves summary details of all repairs from **Repairs file, Estimate file** and associated customer details from Customers File and displays on UI  **Step 3:** System calculates the total repairs cost and displays on the UI.  **Step 5:** System generates a printer-friendly version of the repair report, and sends it to the printer.  **Step 6:** System displays a confirmation message. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
|  |  |  | |
|  | | |
| **Conclusions** | * Repair report is generated. | | |
| **Post conditions** |  | | |
| **Business Rules** |  | | |
| **Implementation Constraints** |  | | |

## **Manage Admin**

This module is responsible for listing jobs status, income analysis, recording customer payments and issuing company invoices.

### List Repair Jobs

#### UC Diagram to generate a list of jobs and their status.

Manager

<<includes>>

#### UC Narrative to generate a list of jobs and their status.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **List Repair Jobs** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.4.1 | | **Date: 24/10/2014** |
| **Priority** | Medium | | |
| **Source** | Staff | | |
| **Primary Business Actor** | Manager | | |
| **Other Participating Actors** |  | | |
| **Description** | Manager wishes to access the status of all the active jobs. This requires the system to generate a list of all the pending repairs. | | |
| **Preconditions** | * Repairs file should exist in the system. | | |
| **Trigger** | * Shop manager wants to list all the jobs. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager requests a list of repair jobs.  **Step 3:** The manager chooses **display by** options:   * **Customer** * **Description** * **Device** * **Total** * **Validated** * **Fixing** * **Resolved** | **Step 2:** The system retrieves details of all the ‘Validated/Fixing/Resolved’ repairs from the **Repairs file, Estimates file** and associated customer details from Customers File and displays them on the interface (UI).  **Step 4:** The system generates a list according to the option chosen by the manager and displays on the system.  **Step 5:** System allows the repairs to be set to Fixing if the ‘Validated’ status is chosen. System also allows the repairs status to be set to Resolved if the ‘Fixing’ status is chosen. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **Set status to Resolved** | **Step 5:** Manager chose sort by repair status to ‘Fixing’  **Step 8:** Manager/staff chooses the repairs to be set to resolved. | **Step 6:** System displays all the ‘Fixing’ repairs and displays on the UI.  **Step 7:** System allows the status to be changed to ‘Resolved’.  **Step 9:** System updates the repair status to ‘Resolved’ and saves it to the repairs file.  **Step 10:** System displays confirmation message. | |
| **Set status to fixing** | **Step 5:** Manager chose sort by repair status to ‘Validated’  **Step 8:** Manager/staff chooses the repairs to be set to fixing. | **Step 6:** System displays all the ‘Validated’ repairs and displays on the UI.  **Step 7:** System allows the status to be changed to ‘Fixing’.  **Step 9:** System updates the repair status to ‘Fixing’ and saves it to the repairs file.  **Step 10:** System displays confirmation message. | |
| **Print repairs listing** | **Step 4:** The manager requests a printed copy of the jobs list. | **Step 5:** The system generates a printed list based on the retrieved data and sends to the default print device.  **Step 7:** The system displays a confirmation message. | |
|  | **Jobs list is printed.** | | |
| **Conclusions** | * A listing of all the repair jobs is generated/printed. | | |
| **Post conditions** | * Details of the jobs displayed cannot be amended or deleted. | | |
| **Business Rules** | * Jobs details may only be viewed, not changed or deleted, when displayed in this way. * The list should include only current jobs. | | |

### Issue Invoice

#### UC Diagram to generate invoice at any given time.

<<includes>>

Manager

Customer

<<includes>>

<<extends>>

<<includes>>

<<includes>>

<<includes>>

#### UC Narrative to generate invoice at any given time.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Issue Invoice** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.4.2 | | **Date: 24/10/2014** |
| **Priority** | Medium | | |
| **Source** | Staff | | |
| **Primary Business Actor** | Manager | | |
| **Other Participating Actors** | Customer | | |
| **Description** | * Customer would like to have a new invoice. | | |
| **Preconditions** | * Customer requests invoice | | |
| **Trigger** | * Repairs file | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** Customer requests a new invoice.  **Step 3:** The staff/manager selects required repair from the list displayed.  **Step 5:** Manager/staff issues a new invoice. | **Step 2:** System retrieves summary details of all ‘**Paid**’ repairs from **Repairs file** & **Estimate file** and associated customer details from **Customers File** and displays on UI  **Step 4:** System retrieves all details of selected repair/customer and displays on UI  **Step 6:** System generates a printer-friendly invoice and sends it to the default printer.  **Step 7:** A confirmation message is displayed. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
|  |  |  | |
| **Conclusions** | * A new invoice is issued. | | |
| **Post conditions** |  | | |
| **Business Rules** |  | | |
| **Implementation Constraints** |  | | |

### Income Analysis

#### UC Diagram to generate income analysis to required specification.

Manager

<<includes>>

<<includes>>

<<includes>>

<<requires>>

#### UC Narrative to generate income analysis to required specification.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Income Analysis** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.4.3 | | **Date: 14/10/2014** |
| **Priority** | Medium | | |
| **Source** | Staff | | |
| **Primary Business Actor** | Manager | | |
| **Other Participating Actors** |  | | |
| **Description** | * Shop manager wishes to access income analysis of a given period. | | |
| **Preconditions** |  | | |
| **Trigger** | * Reporting period | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager requires an income analysis report | **Step 4:** The system displays a 12-monthly report of all the repairs vs total earnings recorded for current year from **Repairs file** and **Estimates** and displays on the UI. | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
|  |  |  | |
|  | **Income Analysis is generated.** | | |
| **Conclusions** |  | | |
| **Post conditions** |  | | |
| **Business Rules** |  | | |
|  |  | | |

### Inventory

#### UC Diagram to allow the system to generate pre-defined repair rates.

Manager

#### UC Narrative to allow the system to generate pre-defined repair rates.

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Inventory** | | **Author: Nazmul Alam** |
| **Use Case Id** | 3.4.4 | | **Date: 24/10/2014** |
| **Priority** | Medium | | |
| **Source** | Staff | | |
| **Primary Business Actor** | Manager | | |
| **Other Participating Actors** |  | | |
| **Description** | * Manager wishes to set pre-defined repair rates. | | |
| **Preconditions** |  | | |
| **Trigger** | * Shop manager wants to set pre-define repair rates. | | |
| **Typical Scenario** | **Actor Action** | **System Response** | |
|  | **Step 1:** The manager/staff wants to add a new part to the inventory.  **Step 3:** Manager enters data:   * **Product Name** * **Description** * **Manufacturer** * **Condition {New, Used, Refurbished}** * **Quantity** * **Price** | **Step 2:** System displays inventory form on the screen.  **Step 4:** System validates data entry   * All entries are mandatory. * Product Name should only contain:   + Letters/numbers   + No trailing spaces   + Between 1-40 characters   + Only one space between every words * Description should only contain:   + No trailing spaces   + Between 1-500 characters   + Only one space between every words * Brand should only contain:   + Letters   + No trailing spaces   + Between 1-50 characters   + Only one space between every words * Quantity should be 1-5 characters and Integer values. * Price should be floating numbers of 6,2 format.   **Step 5:** System generates increment based **Part\_ID** and assigns it to the new part.  **Step 6:** The system saves the new part details in the **Parts file.**  **Step 7:** The system displays a confirmation message | |
| **Alternate Scenarios** | **Actor Action** | **System Response** | |
| **List of Parts** | **Step 7:** Manager/staff wants to see a inventory listing.  **Step 9:** Manager/staff wants to print the parts listing. | **Step 8:** System retrieves all the parts details from Parts file and displays on the UI.  **Step 10:** System generates a printer friendly parts listing and sends it to the default printer.  **Step 11:** System displays a confirmation message. | |
| **Amend part details** | **Step 8:** Manager/staff wants to amend parts listing.  **Step 9:** Manager/staff chooses the part to be amended.  **Step 12:** Manager/staff amends part details. | **Step 10:** System makes sure only one part is selected.  **Step 11:** System displays the amend form. | |
| **Delete parts** | **Step 8:** Manager/staff wants to delete parts listing. | **Step 10:** System checks the parts to be deleted.  **Step 11:** Sysetm delets the selected parts and displays a confirmation message on the UI. | |
| **Invalid Data** | **Step 6:** Manager re-enters data. | **Step 4:** A required field is not entered and/or invalid data type.  **Step 5:** The system displays an appropriate error message and notifies the error sources | |
| **Repair rate is set.** | | |
| **Conclusions** | * New repair rate is set in the system. | | |
| **Post conditions** |  | | |
| **Business Rules** |  | | |
| **Implementation Constraints** |  | | |

# **System Model**

## **Data Flow Diagrams**

**PC Solutions** is considering implementing a computerised system to handle their staffs, customers, repairs, inventory and account records. The following is a description of the system:

Details of all the **Staff**, including managers, interns and normal staff, is recorded in the **Staff File.** Each staff is identified by a unique staff ID and has staff details – surname, forename, DOB, gender, address1, address2, town, county, telephone, mobile, e-mail, staff type and status. The company has the ability to Add, Amend and List the details of all the staff in the Staff File.

The company will record every customer before they can proceed with a new repair, therefore, the company needs to store all the **Customers** in the **Customers File**. Each customer is identified by a unique customer ID and has the details – Surname, Forename, DOB, Gender, Address1, Address2, Town, County, Telephone, Mobile, E-Mail, Customer Type and Status. The company has the ability to Add, Amend, Query and List the details of all the customers in the Customers File.

The requirements of an incoming repair job are logged in the **Repairs File.** Each job is assigned a job reference number and may be for a new customer or an existing customer. For new customers, the customer details are recorded in the Customer File. A job is for only one ‘Repair ID’ but a customer may have several jobs on-going at the same time.

### Level 0 DFD

Repair Details

Customer Details

**CUSTOMER**

**Computer Repair System**

### Level 1 DFD

Estimate Parts Details

Staff Details

Estimate Details

Repair Details

Repair Details

Customer Details

Customer Details

P2

Manage Customers

Customer Details

Customer

P1

Manage Staff

P4

Manage Admin

P3

Manage Repairs

D1

Staff File

D2

Customer File

D3

Repairs File

D4

Estimate File

D5

EstParts File

D6

Parts File

### Level 2 DFD (Process P1 – Manage Staff)

Staff Details

Retrieve Staff Details

Staff Details

New Staff Details

Get Staff Details

P1.1

New

Staff

P1.2

Amend Staff

P1.3

List Staff

D1

Staff File

### Level 2 DFD (Process P2 – Manage Customers)

New Customer Details

Customer Details

Repair Details

Customer Details

Customer Details

P2.3

Query Customer

Customer

Repairs File

D3

Estimate File

D4

Customer

Estimate Details

Customer Details

Customer Details

Customer Details

New Customer Details

Get Customer Details

P2.1

New

Customer

P2.2

Amend Customer

P2.4

List

Customer

D1

Customer File

### Level 2 DFD (Process P3 – Manage Repairs)

Customer Details

P3.6

Repair

Report

Parts Details

Repair Details

Repairs File

D3

Estimate Parts Details

Estimate Parts Details

EstParts File

D5

Estimate Details

Repair Details

Estimate Details

Estimate Details

Estimate File

D4

Parts Details

EstParts File

D5

Customer Details

Estimate Details

Estimate Details

Estimate File

D4

Customer Details

Repair Details

Repair Details

Repair Details

Repair Details

Repair Details

Repair Details

Customer Details

Customer Details

Repair validation

Repair

Customer Details

Customer Details

P3.4

Finalise Repair

Customer File

D2

Customer File

D2

Customer

P3.3

Validate Repair

Repairs File

D3

P3.1

New

Repair

P3.2

New

Estimate

P3.5

Collect

Repair

Customer

Parts File

D6

### Level 2 DFD (Process P4 – Manage Admin)

Job Details

Estimate Details

EstParts File

D5

Repairs File

D3

EstPart Details

Customer Details

Invoice

Rates Details

Job Details

P4.4

Inventory

Parts File

D6

Repairs File

D3

Estimate File

D4

Customer File

D2

Customer

Repair Details

Customer Details

P4.1

List Repair Jobs

P4.2

Issue Invoice

P4.3

Income Analysis

Cost Details

Estimate Details

# **Data Model**

## **UML Class Diagram**

Rep\_ID {pk}

Description

Creation\_Date

Collection\_Date

Problem\_Type

Status

**REPAIR**

Part\_ID {pk}

Description

Shop\_Price

Condition

Vendor

**PART**

Est\_ID {pk}

Estimate\_Date

Total

**ESTIMATE**

0..\*

0..\*

0..\*

0..\*

Staff\_ID {pk}

Surname

Forename

DOB

Address

Tel

Email

Staff\_Type

Status

**STAFF**

Cust\_ID {pk}

Surname

Forename

DOB

Address

Tel

Email

Cust\_Type

Status

**CUSTOMER**

1

1

1

1

1

1

Quantiy

Retail\_Price

**EstPart**

has

includes

has

manages

belongs to

# **Database Schema**

## **Relational Schema**

**Customers**(Cust\_ID, Surname, Forename, DOB, Address1, Address2, Town, County, Telephone, Email, Cust\_Type, RegDate, Status)

**Staff(**Staff\_ID, Surname, Forename, DOB, Address1, Address2, Town, County, Telephone, Email, Staff\_Type, RegDate, Status, username, password)

**Repairs(**Rep\_ID, *Cust\_ID*, *Staff\_ID*, Description, DeviceType, Brand, Model, OS, RepairDate, CollectionDate, Status)

**Estimates(**Rep\_ID, Staff\_ID EstimateDate, Status, VAT, Discount, Total)

**Parts(**Part\_ID, Part\_Name, Description, Vendor, Condition, TotalStock, ShopPrice)

**EstParts(***Rep\_ID, Part\_ID*, Quantity, RetailPrice)

## **Schema: CompSys**

### Relation: Customers

Cust\_ID number(9)

Surname varchar(50) NOT NULL

Forename varchar(100) NOT NULL

DOB date NOT NULL

Address1 varchar(100) NOT NULL

Address2 varchar(100)

Town varchar(30) NOT NULL

County varchar(9) NOT NULL

Telephone char(15) NOT NULL

Email varchar(500) NOT NULL

Cust\_Type varchar(20) CHECK (Cust\_Type IN ('Normal', 'Student', 'OAP')) NOT NULL

RegDate date DEFAULT SYSDATE

Status char(20) DEFAULT 'A'

PRIMARY KEY Cust\_ID

UNIQUE Email, Telephone

### Relation: Staff

Staff\_ID numeric(9)

Surname varchar(50) NOT NULL

Forename varchar(100) NOT NULL

DOB date NOT NULL

Address1 varchar(100) NOT NULL

Address2 varchar(100)

Town varchar(30) NOT NULL

County varchar(9) NOT NULL

Telephone char(15) NOT NULL

Email varchar(500) NOT NULL,

Staff\_Type varchar(20) CHECK (Staff\_Type IN ('Manager', 'Staff', 'Intern')) NOT NULL

RegDate date DEFAULT SYSDATE

Status char(20) DEFAULT 'A'

username char(20) NOT NULL

password char(20) NOT NULL

PRIMARY KEY Staff\_ID

UNIQUE username, Email, Telephone

### Relation: Repairs

Rep\_ID number(9)

Cust\_ID number(9) NOT NULL

Staff\_ID number(9) NOT NULL

Description varchar(1000) NOT NULL

DeviceType char(20) CHECK (DeviceType IN ('Laptop', 'Desktop', 'Printer', 'Other')) NOT NULL

Brand varchar(50) NOT NULL

Model varchar(50) NOT NULL

OS varchar(20) CHECK (OS IN ('Windows 7', 'Windows 8', 'Windows Vista', 'Windows Older', 'MacOS X', 'Linux', ‘Other’)) NOT NULL

RepairDate date DEFAULT SYSDATE

CollectionDate date

Status varchar(30) DEFAULT ‘New’ CHECK (Status IN ('New', ‘Estimated’, 'Validated', 'Fixing’, ‘Resolved’, 'Invoiced', ‘Paid’)) NOT NULL

Primary Key Rep\_ID

FOREIGN KEY Cust\_ID REFERENCES Customers

FOREIGN KEY Staff\_ID REFERENCES Staff

### Relation: Estimates

Rep\_ID number(9)

Staff\_ID number(9) NOT NULL

EstimateDate date DEFAULT SYSDATE

VAT decimal(4,2)

Discount decimal(4,2),

Total decimal(9,2) NOT NULL

PRIMARY KEY Rep\_ID

FOREIGN KEY Staff\_ID REFERENCES Staff

FOREIGN KEY Rep\_ID REFERENCES Repairs

### Relation: Parts

Part\_ID number(9)

Part\_Name varchar(40) NOT NULL

Description varchar(500) NOT NULL

Vendor varchar(50) NOT NULL

Condition varchar(11) CHECK (Condition IN ('New', 'Refurbished', 'Used'))

TotalStock number(5) NOT NULL

ShopPrice decimal(6,2) NOT NULL

PRIMARY KEY Part\_ID

### Relation: EstParts

Rep\_ID number(9)

Part\_ID number(9) NOT NULL

Quantity number(4) NOT NULL

RetailPrice decimal(6,2) NOT NULL

PRIMARY KEY Rep\_ID, Part\_ID

FOREIGN KEY Part\_ID REFERENCES Parts

FOREIGN KEY Rep\_ID REFERENCES Estimates

# **Conclusion**

CompSys (computer repair system) is a custom application written from the ground up for managing a small computer store and repair shop. It has been used and refined on a daily basis for over 8 years now in our own small shop.

**References:**

<http://www.academia.edu/3046773/Requirement_Analysis_Document_for_Recruitment_Management_System>