David Pyle

041110777@my.central.edu.au

Assignment 3

UI .NET

Contents

[1 Introduction 3](#_Toc449426889)

[2 Prototype Tool 3](#_Toc449426890)

[3 Client Requirements 3](#_Toc449426891)

[3.1 Login 3](#_Toc449426892)

[3.2 Customer Details 3](#_Toc449426893)

[3.3 Invoice Details 3](#_Toc449426894)

[3.4 Search 3](#_Toc449426895)

[3.5 Edit 3](#_Toc449426896)

[3.6 Change Interface Background Image 3](#_Toc449426897)

[3.7 Interface 3](#_Toc449426898)

[4 UI Design and Components 4](#_Toc449426899)

[4.1 Overview 4](#_Toc449426900)

[4.2 Login Form 4](#_Toc449426901)

[4.2.1 Login Error 4](#_Toc449426902)

[4.3 Customer Details 5](#_Toc449426903)

[4.3.1 View Customer Details 5](#_Toc449426904)

[4.3.2 Add a Customer 6](#_Toc449426905)

[4.3.3 Edit Customer 8](#_Toc449426906)

[4.3.4 Delete Customer 8](#_Toc449426907)

[4.4 Invoice Details 9](#_Toc449426908)

[4.4.1 View Invoice Details 9](#_Toc449426909)

[4.4.2 Add Invoice 10](#_Toc449426910)

[4.4.3 Edit Invoice 11](#_Toc449426911)

[4.4.4 Delete Invoice Item 12](#_Toc449426912)

[4.4.5 Delete Invoice 12](#_Toc449426913)

[4.5 Search 13](#_Toc449426914)

[4.5.1 Search by Customer Name 13](#_Toc449426915)

[4.5.2 Search by Invoice Number 13](#_Toc449426916)

[4.6 Interface 15](#_Toc449426917)

[4.6.1 Change Interface Background Image 15](#_Toc449426918)

[5 Testing 16](#_Toc449426919)

[5.1 Login Test 16](#_Toc449426920)

[5.2 View Customer Test 16](#_Toc449426921)

[5.3 View Invoices Test 16](#_Toc449426922)

[5.4 Add Customer Test 17](#_Toc449426923)

[5.5 Add Invoice Test 18](#_Toc449426924)

[5.6 Edit Invoice Test 18](#_Toc449426925)

[5.7 Delete Invoice Test 19](#_Toc449426926)

[5.8 Edit Customer Test 20](#_Toc449426927)

[5.9 Delete Customer Test 20](#_Toc449426928)

[5.10 Search Customers Test 21](#_Toc449426929)

[5.11 Search Invoices Test 21](#_Toc449426930)

[5.12 Change Interface Background Test 22](#_Toc449426931)

[6 Annex A – UI Component Details 23](#_Toc449426932)

[6.1 Login Form UI Components 23](#_Toc449426933)

[6.2 Main Form UI Components 23](#_Toc449426934)

[6.3 Add Customer, Edit Customer UI Components 24](#_Toc449426935)

[6.4 Add Invoice, Edit Invoice UI Components 25](#_Toc449426936)

# Introduction

This report will describe the design for the ABC Invoice application. The purpose of the application is to help manage customer and invoice details.

The report documents the client requirements, provides descriptions of the user interface (UI) components used, screenshots of the proposed design and a test table that can be used for acceptance testing.

# Prototype Tool

The UI was designed using Visual Studio 2013 and the application is written in C#.

# Client Requirements

The client requirements for the ABC Invoice application are listed below in sections 3.1 to 3.9.

## Login

**Requirement**: The system must provide a login screen. Access to the main application is only granted once the user has been authenticated. If authentication fails an error message is displayed.

## Customer Details

**Requirement:** The system must be able to take in multiple customer records containing the following customer details:

* Customer Number
* Customer Name
* Address
* Invoice number
* Contact number

## Invoice Details

**Requirement**: The system must be able to take in multiple invoice records containing the following invoice details:

* Invoice Number
* Customer Number
* Description
* Costs
* Payment Date

## Search

**Requirement**: The system must be able to search for records by:

* Invoice Number
* Customer Name

## Edit

**Requirement**: The system must be able to edit customer and invoice details.

## Change Interface Background Image

**Requirement**: The system must provide the facility to change the background image of the application’s interface.

## Interface

**Requirement**: In addition, the interface design to be simple, sleek and consistent.

## Data Storage

**Requirement:** All data from the application must be stored in a database. When the application is opened data from the database must be loaded for the user to interact with and saved to the database when changes are made.

## External Library

**Requirement:** The application must make use of an external library (DLL file)

# Design Overview

The design of the application is primarily based around the customer entity. The customer is central to how a user interacts with the application. From the main view the user can view, add, edit, delete and search customer records as well as view, add, edit and delete invoices for a customer.

This design approach avoids the possibility of orphan invoice records, i.e. an invoice that doesn’t belong to a customer. The design also simplifies how a user can view and interact with the customer and invoice data. By only having a single main view based around the customer it avoids switching between different ways of viewing the data. i.e. searching for customers or invoices can all be done from the main form, there is no need to ‘switch’ between a customer view or invoice view.

The ABC Invoice application supports the creation of an unlimited number of customers and invoices. Invoices can contain an unlimited number of items.

Application data is stored in a relational database. After successful authentication, a connection to the database is established and any existing data is read and displayed on the main interface. As the user interacts with the data any changes are reflected in the database. The design of the database can be found in Annex B.

The external library **CustInvoiceLogin.dll** is used to retrieve the application’s username and password during login authentication. It contains two methods; getUsername() and getPassword().

*\*Unlimited = maximum storage of database*

## UI Design and Components

The following sections detail the components that make up the user interface (UI) and lists the client requirements addressed by the

## Login Form

This is the first form displayed when the application opens.

Section 6.1 in Annex A describes each UI component for the Login form.

**Requirements addressed**: 3.1, 3.7, 3.9

**Input:** username, password

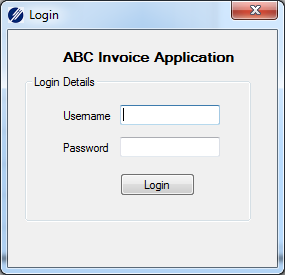


Figure 4.1 Login Screen

### Login Error

If authentication fails an error message is displayed.

**Input:** username, password

**Output:** error message

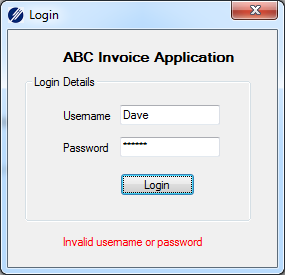


Figure 3.2 Login Screen error

## Customer Details

The ability to view, add, edit and delete customer records.

**Requirements addressed:** 3.2, 3.4, 3.5, 3.7, 3.8

Customer records are stored in a database and read into a data table when the application opens. When adding or editing customer records the same customer form is used along with all of its UI components (buttons, labels etc.). In the case of adding a new record all fields are blank, but, if editing, the fields are populated with the selected customer details.

The database is updated whenever the user saves a customer record either by editing an existing record or adding a new customer.

Section 6.2 and 6.3 in Annex A describes each UI component for the main form and customer form.

### View Customer Details

**Output**: customer number, company, first name, last name, street address, suburb, state, post code, contact number and any associated invoices.

The main form in figure 4.3.1 shows how the customer details are displayed (highlighted by red box).

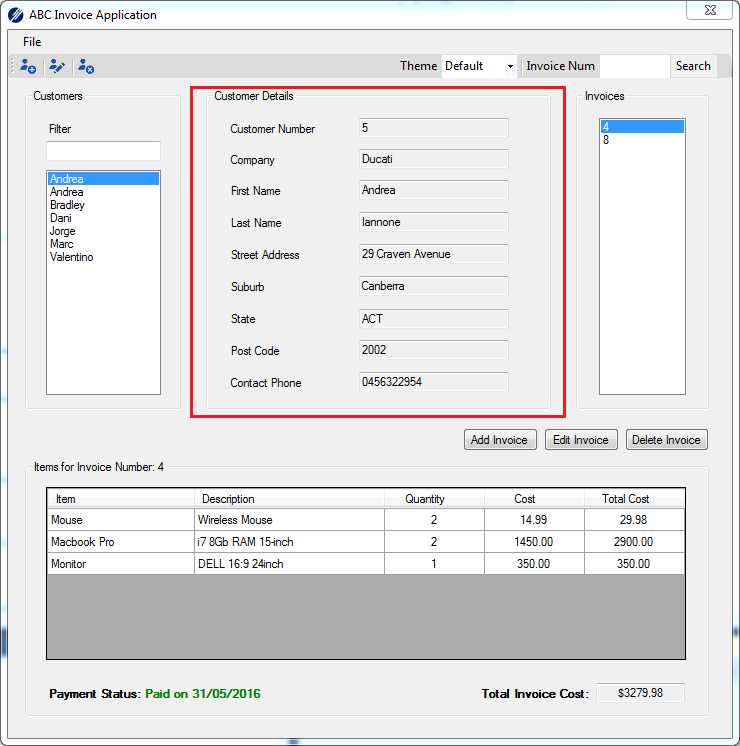


Figure 4.3.1 Customer Details

### Add a Customer

**Input:** customer number\*, company, first name, last name, street address, suburb, state, post code, contact number

\*When adding a new customer, a new customer number will be auto-generated and returned by the database via a stored procedure.

The add customer form can be accessed by keyboard shortcut (Ctrl-N), File menu ‘Add Customer’ or the ‘Add Customer’ button.

All fields are required for a customer to be created.

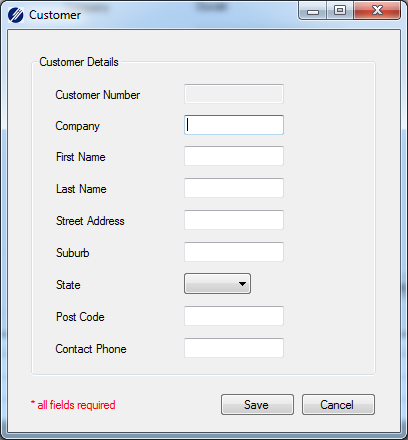


Figure 4.3.2 Add Customer

#### Add a Customer Error

**Input:** customer number\*, company, first name, last name, street address, suburb, state, post code, contact number

\*When adding a new customer, a new customer number will be auto-generated and returned by the database via a stored procedure.

**Output:** error message

If the user attempts to save the record without completing all fields an error message will be displayed. When the ‘Save’ button is clicked the customer record is added to the customer data table and database.

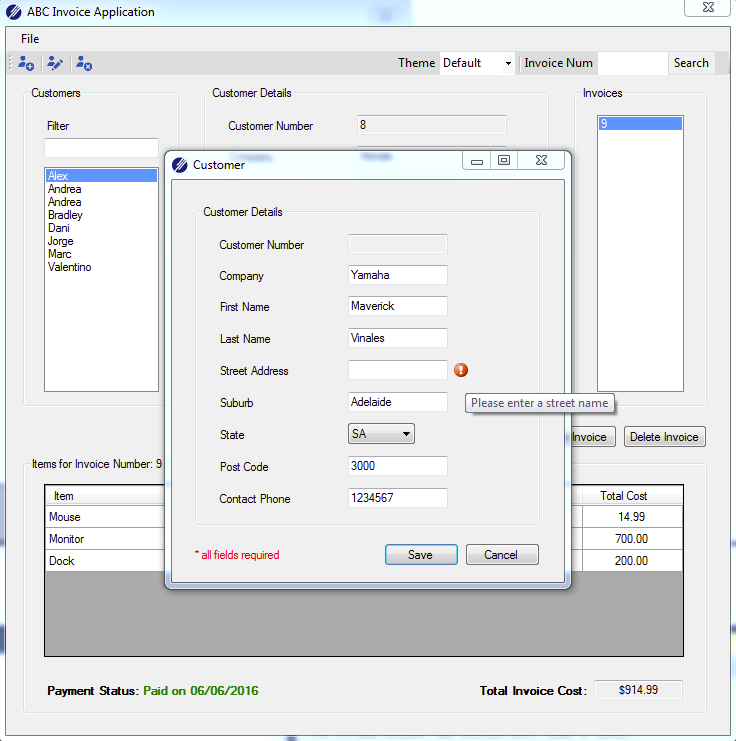


Figure 4.3.2.1 Add Customer Error

### Edit Customer

**Input**: customer number\*, company, first name, last name, street address, suburb, state, post code, contact number

\*Customer number will be read only

The edit customer form can be accessed by keyboard shortcut (Ctrl-E), File menu ‘Edit Customer’ or the ‘Edit Customer’ button.

All fields are required for a customer when saving the changes. When the ‘Save’ button is clicked the customer record is updated in the customer data table and database.

When editing a customer, the same customer form is used as in figures 4.3.2 and 4.3.2.1.

### Delete Customer

**Input:** customer name

**Output:** confirmation dialog

Deleting a customer can be done by keyboard shortcut (Ctrl-D), File menu ‘Delete Customer’ or the Delete Customer button.

A confirmation dialog is displayed when deleting a customer. If the customer has any associated invoices the dialog message will notify the user that these records will also be deleted.

If ‘Yes’ is selected on the Delete Confirmation dialog the customer and any associated invoice records are removed from the customer data table and database.

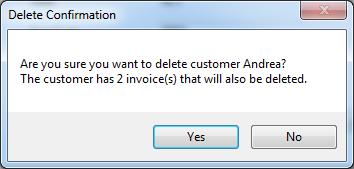


Figure 4.5 Delete Customer Confirmation Dialog

## Invoice Details

The ability to view, add, edit and delete invoice records.

**Requirements addressed:** 3.3, 3.4, 3.5, 3.7, 3.8

Invoice records are stored in a database and read into a data table when the application opens. The invoice data table is related to the customer data table within the dataset via a data relation. This sets up a master-detail relationship between the two tables.

When adding, or editing invoice records the same invoice form is used along with all of its UI components. In the case of adding a new record all fields are blank but if editing, the fields are populated with the selected invoice details.

Invoice items are also stored in the database and read into the items data table. The items data table is related to the invoice data table in a similar manner to the customer-invoice relationship.

Section 6.2 and 6.4 in Annex A describes each UI component for the main form and invoice form.

### View Invoice Details

**Output:** invoice number, customer details, payment due date, payment status, invoice total cost and for each item on the invoice – item number, item name, item description, item cost, quantity and total item cost

The main form in figure 4.6.1 shows how the invoice details are displayed (highlighted by red box).

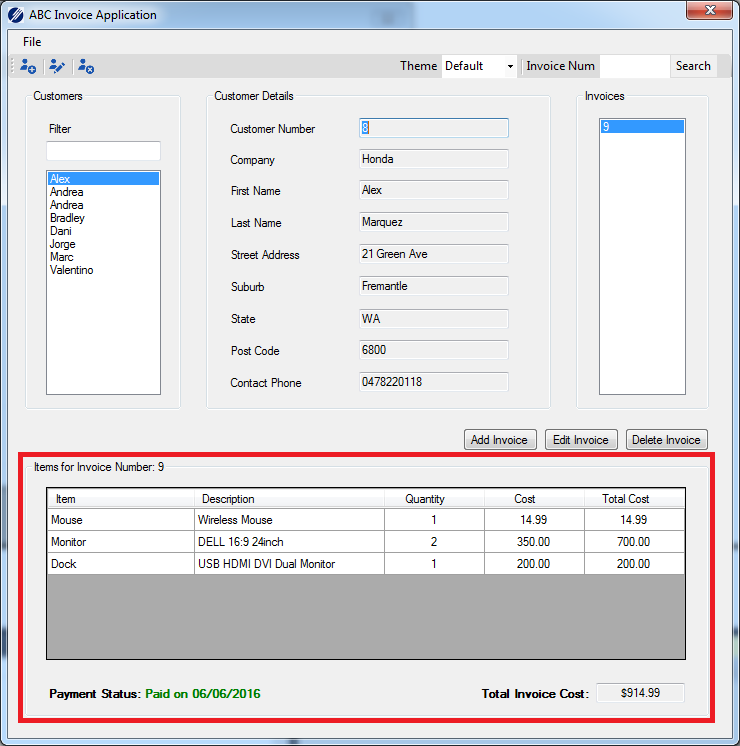


Figure 4.6.1 Invoice Details

### Add Invoice

**Input:** invoice number, payment due date, payment status and at least one item by selecting item name and quantity.

The item’s number, description, cost and total item cost will be filled in automatically by the system.

\*When adding a new invoice, a new invoice number will be auto-generated and returned by the database via a stored procedure.

The add invoice form can be accessed by the Add Invoice button. It is not possible to save an invoice (Save button is disabled) unless at least one item has been added.

Items can be added by selecting an item name from the drop-down list. Once an item is added the item number, description, cost, quantity (initially set to 1) and total cost are automatically filled in. The user can then edit the quantity for the item. Once an item has been added it is removed from the list of items available in order to avoid duplicate entries in the database. If an item is removed from the invoice, then it is added back to the list of available items.

When the ‘Save’ button is clicked the main invoice details (Payment Due Date, Payment Status and Payment Date) are added to the invoice data table and database. The database will return a new invoice number which is then used to add the invoice items to the invoice items data table and then the database.

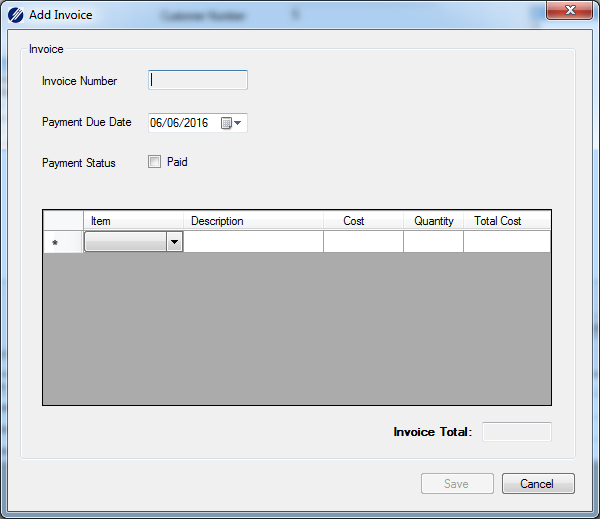


Figure 4.6.2 Add Invoice

### Edit Invoice

**Input**: payment due date, payment status, item name, quantity

When editing an item can be added by selecting an item name from the drop-down list. Once an item is added the item number, description, cost, quantity (initially set to 1) and total cost are automatically filled in. The user can then edit the quantity for the item. Once an item has been added it is removed from the list of items available in order to avoid duplicate entries in the database. If an item is removed from the invoice, then it is added back to the list of available items.

When the ‘Save’ button is clicked the main invoice details (Payment Due Date, Payment Status and Payment Date) are updated in the invoice data table and database. Next, any records for the edited invoice are deleted from the invoice-items database table before the new invoice are inserted. This provides an efficient way to update the invoice-items table.

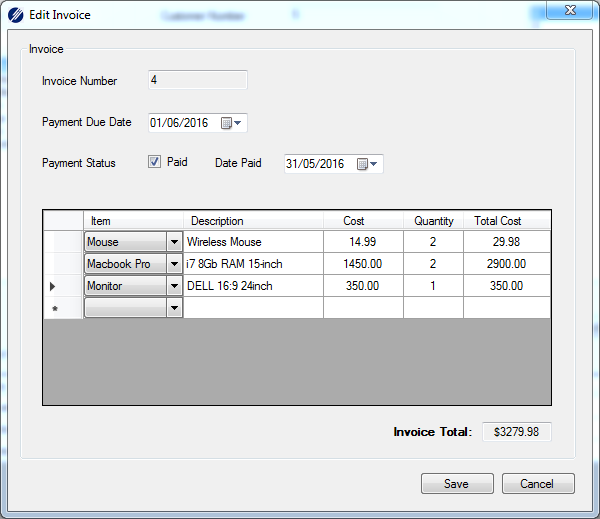


Figure 4.6.3 Edit Invoice

### Delete Invoice Item

The ability to delete an item from an invoice.

Item rows can be deleted by selected the row and pressing the delete key. If all items are deleted the ‘Save’ button is disabled.

The item is added back to the list of available items to add if deleted.

### Delete Invoice

**Input:** invoice number

**Output:** confirmation dialog

Deleting an invoice is done by selecting an invoice number from the list of invoices and then using the ‘Delete Invoice’ button.

A confirmation dialog is displayed when deleting an invoice. If ‘Yes’ is selected the invoice is removed from the customer data table and database.

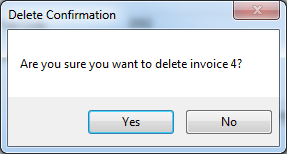


Figure 4.6.5 Delete Invoice Confirmation Dialog

.

## Search

The ability to search customer records by customer name and to search for an invoice by invoice number.

**Requirements addressed:** 3.4, 3.7

Every customer name is displayed in the customer list and can be searched by using the ‘Filter’ field. Invoice numbers related to a customer are displayed in the invoice list box and can be searched by using the ‘Invoices’ search field and ‘Search’ button.

### Search by Customer Name

The ability to search for a customer record by customer name

**Input**: customer name

Figure 4.7.1 shows how customer records can be searched for by entering text into the Filter field. As text is entered, the customer list is filtered to only those customer names that match the filter text.

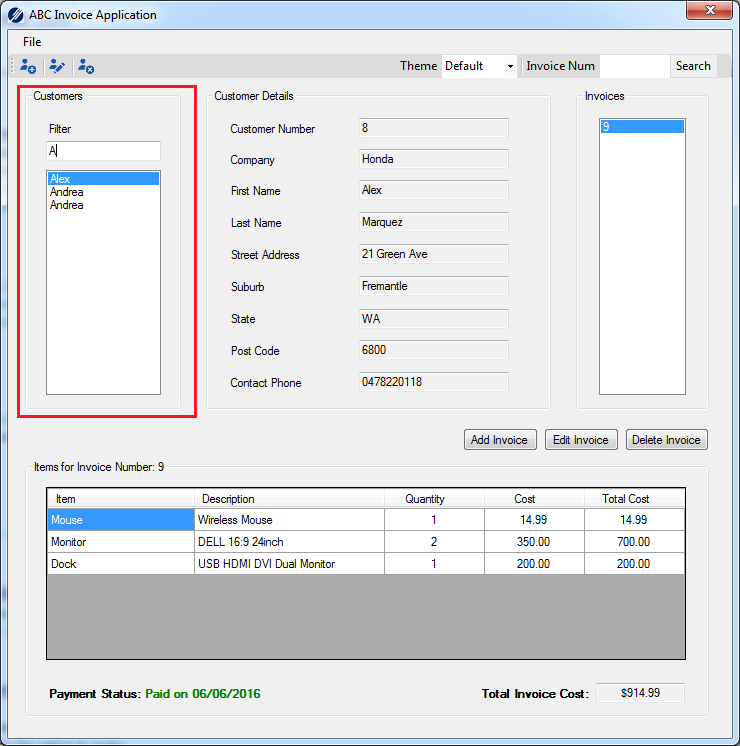


Figure 4.7.1 Customer Name Search

### Search by Invoice Number

The ability to search for an invoice by invoice number

**Input**: invoice number

Figure 4.7.2 shows how invoice records can be searched by entering text into the ‘Invoices’ search field and pressing the ‘Search’ button. To retrieve the details for the invoice the ‘Search’ button is clicked.

Only numbers are allowed to be entered into the search field. Non-numerical characters are disabled from being entered into the search field.

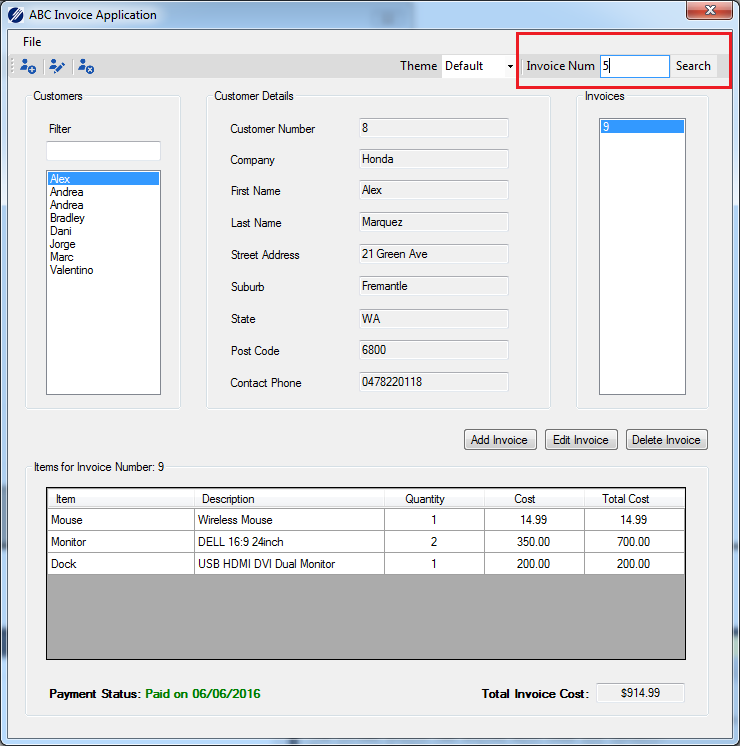


Figure 4.7.2 Invoice Number Search

#### Search by Invoice Number Message Dialog

**Input**: invoice number

**Output:** search results message dialog

If an invoice number search is performed and no records are found a message is displayed to the user.

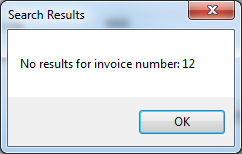


Figure 4.7.2.1 No Results Message Dialog

## Interface

The UI has been designed using a consistent approach to layout, style and colours for the various forms that make up the application. The user also has the ability to select between three ‘themes’; default, dark or light.

Each theme changes the background image of the interface.

**Requirements addressed:** 3.6, 3.7

### Change Interface Background Image

**Input**: theme name

The theme can be changed by selecting from the ‘Theme’ drop down list on the toolbar as shown in figure 4.8.1.

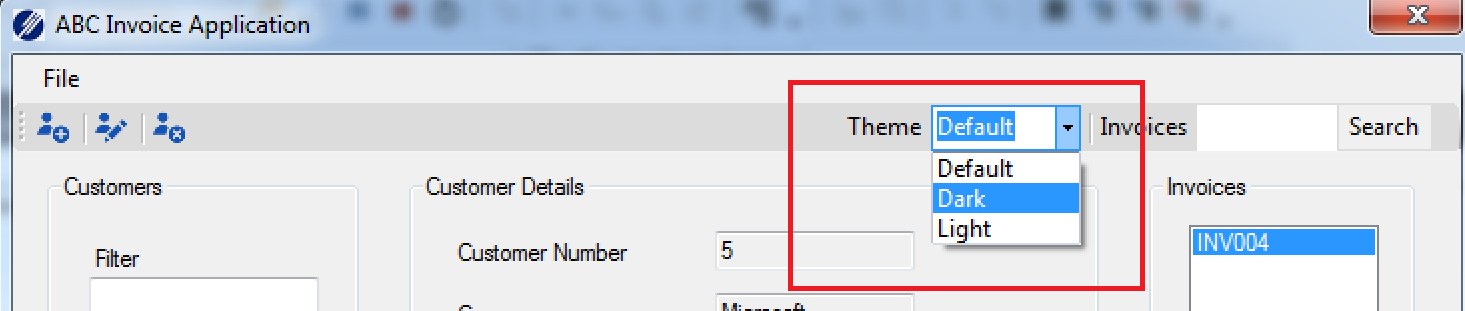


Figure 4.8.1 Theme Selection

Figures 4.8.2 shows the look and feel of each of the three themes.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Figure 4.8.2 Theme Look and Feel (default, dark and light)

# Database Design

Figure 5.1 shows the design for the relational database used for the ABC Invoice application. There are 4 tables; Customer, Invoice, Item and InvoiceItem.

Customer has a one-to-many relationship with Invoice and is linked by the CustNum field.

Invoice and Item have a many-to-many relationship and so use the linking table InvoiceItem to implement the relationship. The primary key of InvoiceItem is made up of the primary keys of Invoice and Item (InvoiceNum and ItemNum).

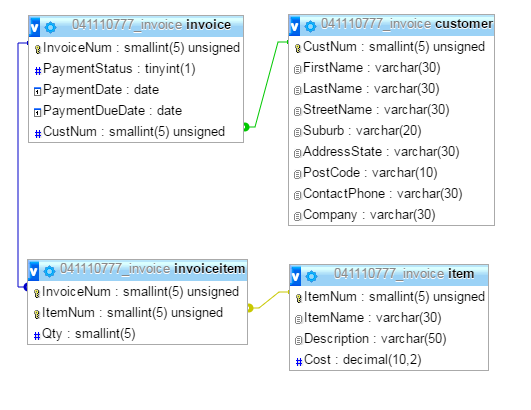


Figure 5.1 Database design for ABC Invoice application

## Referential Integrity

Customer->Invoice: delete cascade, update cascade

Invoice->InvoiceItem: delete cascade, update cascade

Item->InvoiceItem: delete cascade, update cascade

The above referential integrity rules means that if a customer record is deleted/updated then any associated invoices for that customer will also be deleted/updated.

Similarly, if an invoice or item is deleted/updated then the associated record in the InvoiceItem table will also be deleted/updated.

The Data Definition Language (DDL) for the ABC Invoice application database and sample data can be found at Annex B.

## Stored Procedures

The ABC Invoice application allows the creation of new customer and invoice records. These records are saved to the dataset within the application and also to the database. When a new record is created the database automatically creates a new id for the record. In the case of the Customer table the id is CustNum and for the Invoice table the id is InvoiceNum.

In order to keep the state of the database and dataset synchronised stored procedures are used within the database for the insert queries. The stored procedures return the new ID assigned to the created record back to the application. This means the most up-to-date data is displayed to the user at all times.

# Testing

The following test table describes the details and steps of several tests that were used for checking the functionality of the ABC Invoice application.

## Login Test

In the case of a failed test, comments are to be provided in the Pass/Fail column.

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | Login | | |
| **Description:** | Tests authentication of username and password | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 1.0 | Open application.  Type ‘MyFitness’ in username.  Type ‘pass’ in password.  Click ‘Login’ button. | Login form disappears  Main form is displayed | Pass |
| 1.1 | Click’ File’ then ‘Exit’.  Open the application.  Type ‘abc’ in username.  Type ‘abc’ in password.  Click ‘Login’ button. | Error message ‘Invalid username or password’ will be displayed | Pass |
| 1.2 | Click red X close button | Application closes | Pass |
| 1.3 | Open application.  Login as per Test 1.0  Click File menu item then ‘Exit’ | Application closes | Pass |

## View Customer Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | View Customers | | |
| **Description:** | Tests that customer details are displayed correctly | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 2.0 | Login (as per Test 1.0) | Main form is displayed:  Customer name ‘Alex’ is selected.  Customer details for ‘Alex’ are displayed:  Customer number (8), company (Honda), first name (Alex), last name (Marquez), street address (21 Green Ave), suburb (Fremantle), state (WA), post code (6800), contact phone (0478220118). | Pass |
| 2.1 | Click on a different customer name in the customer list | The customer name clicked on is selected.  Customer details for the selected customer are displayed in the Customer details section. | Pass |
| 2.2 | Click’ File’ then ‘Exit’. | Application closes. | Pass |

## View Invoices Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | View Invoices | | |
| **Description:** | Tests that invoice details are displayed correctly | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 3.0 | Login (as per Test 1.0) | Main form is displayed.  Customer name ‘Alex’ is selected.  Customer details for ‘Alex’ are displayed:  Customer number (8), company (Honda), first name (Alex), last name (Marquez), street address (21 Green Ave), suburb (Fremantle), state (WA), post code (6800), contact phone (0478220118).  Invoice number 9 is selected.  Invoice Details section contains the following items   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Mouse | Wireless Mouse | 1 | 14.99 | 14.99 | | Monitor | DELL 16:9 24inch | 2 | 350.00 | 700.00 | | Dock | USB HDMI DVI Dual Monitor | 1 | 200.00 | 200.00 |   Payment Status shows ‘Paid on 06/06/2016.’  Total Invoice Cost shows ‘$914.99’ | Pass |
| 3.1 | Click on customer name ‘Valentino’ | Customer name ‘Valentino ’is selected.  Customer details for ‘Valentino’ are displayed:  Customer number (1), company (Yamaha), first name (Valentino), last name (Rossi), street address (46 Grand Prom), suburb (Dianella), state (WA), post code (6059), contact phone (0478220117). | Pass |
| 3.2 | Click on invoice number 2 | Invoice number 2 is selected.  Invoice Details section contains the following items   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Monitor | DELL 16:9 24inch | 1 | 350.00 | 350..00 | | Dock | USB HDMI DVI Dual Monitor | 2 | 200.00 | 400.00 |   Payment Status shows ‘Overdue. Payment was due on 06/06/2016.’  Total Invoice Cost shows ‘$750.00’ | Pass |
| 3.3 | Click’ File’ then ‘Exit’. | Application closes. | Pass |

## Add Customer Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | Add Customer | | |
| **Description:** | Tests that a new customer can be added | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 4.0 | Login (as per Test 1.0)  Click on ‘Add Customer’ button | The Customer form will be displayed.  Customer Number field is read only.  All other fields are empty. | Pass |
| 4.1 | Type ‘ABC’ in company field.  Type ‘Fred’ in first name field.  Type ‘Bloggs’ in last name field.  Type ‘333 Smith St’ in street address field.  Type ‘Perth’ in suburb field.  Select ‘WA’ from state drop down list.  Type ‘6000’ in postcode field.  Click the ‘Save’ button | An error symbol will be displayed next to the ‘Contact Phone’ field. | Pass |
| 4.2 | Type ‘12345678’ into the contact phone field.  Click the ‘Save’ button | The Customer form will disappear.  The new customer’s name will be added and selected in the customer list.  The customer details will appear in the Customer Details section. | Pass |

## Add Invoice Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | Add Invoice | | |
| **Description:** | Tests that a new invoice can be added | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 5.0 | Click on a customer name ‘Fred’ | Customer name ‘Fred’ is selected.  Customer details for ‘Fred’ are displayed:  Customer number (9), company (ABC), first name (Fred), last name (Bloggs), street address (333 Smith St), suburb (Perth), state (WA), post code (6000), contact phone (12345678). | Pass |
| 5.1 | Click on ‘Add Invoice’ button | The ‘Add Invoice’ form will be displayed.  ‘Invoice Number’ field is read only.  Payment Due Date is set to today’s date.  Payment Status checkbox is unchecked.  No items are listed.  The ‘Save’ button is disabled. | Pass |
| 5.2 | Check the ‘Paid’ checkbox next to ‘Payment Status’.  Set payment date to yesterday’s date.  Select ‘Monitor’ from the Item drop-down list.  Change the quantity to 2. | A tick will be displayed inside the ‘Paid’ checkbox.  The Date Paid label and Date Paid date picker will show. Date Paid date picker is set to yesterday’s date.  The Item’s table will show:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Monitor | DELL 16:9 24inch | 2 | 350.00 | 700..00 |   Invoice Total field shows ‘$700.00’  The ‘Save’ button is enabled. | Pass |
| 5.3 | Select ‘Monitor’ from the Item drop-down list. | The item will be added to the item’s table:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Monitor | DELL 16:9 24inch | 2 | 350.00 | 700..00 | | Monitor | DELL 16:9 24inch | 1 | 350.00 | 350..00 |   Invoice Total field shows ‘$1050.00’  The ‘Save’ button is enabled. | Pass |
| 5.4 | Click the ‘Save’ button | The ‘Add Invoice’ form disappears.  The main form is displayed.  Customer name ‘Fred’ is selected in the customer list.  The invoice number 9 is selected in the invoices list.  Invoice Details section contains the following items:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Monitor | DELL 16:9 24inch | 2 | 350.00 | 700..00 | | Monitor | DELL 16:9 24inch | 1 | 350.00 | 350..00 |   Payment Status shows ‘Paid on <yesterday’s date>  Total Invoice Cost shows ‘$1050.00’ | Fail. An exception occurred stating that the InvoiceItem’s table could not be updated due an integrity violation.  See **Section 6.6** for ‘Add Invoice Retest’ after bug fix. |

## Add Invoice Retest

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | Add Invoice Retest | | |
| **Description:** | Tests that a new invoice can be added. | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 5.0a | Click on a customer name ‘Fred’ | Customer name ‘Fred’ is selected.  Customer details for ‘Fred’ are displayed:  Customer number (9), company (ABC), first name (Fred), last name (Bloggs), street address (333 Smith St), suburb (Perth), state (WA), post code (6000), contact phone (12345678). | Pass |
| 5.1a | Click on ‘Add Invoice’ button | The ‘Add Invoice’ form will be displayed.  ‘Invoice Number’ field is read only.  Payment Due Date is set to today’s date.  Payment Status checkbox is unchecked.  No items are listed.  The ‘Save’ button is disabled. | Pass |
| 5.2a | Check the ‘Paid’ checkbox next to ‘Payment Status’.  Set payment date to yesterday’s date.  Select ‘Monitor’ from the Item drop-down list.  Change the quantity to 2. | A tick will be displayed inside the ‘Paid’ checkbox.  The Date Paid label and Date Paid date picker will show. Date Paid date picker is set to yesterday’s date.  The Item’s table will show:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Monitor | DELL 16:9 24inch | 2 | 350.00 | 700..00 |   Invoice Total field shows ‘$700.00’  The ‘Save’ button is enabled.  The monitor item is removed from the drop-down list of items. | Pass |
| 5.3a | Click on the Item drop-down list. ‘Monitor’ is no longer available for selection.  Select ‘Dock’ from the drop down list | The item will be added to the item’s table:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Monitor | DELL 16:9 24inch | 2 | 350.00 | 700..00 | | Dock | USB HDMI DVI Dual Monitor | 1 | 200.00 | 200.00 |   Invoice Total field shows ‘$900.00’  The ‘Save’ button is enabled. | Pass |
| 5.4a | Click on the Item drop-down list. | ‘Dock is no longer available for selection. | Pass |
| 5.5a | Click the ‘Save’ button | The ‘Add Invoice’ form disappears.  The main form is displayed.  Customer name ‘Fred’ is selected in the customer list.  The invoice number 9 is selected in the invoices list.  Invoice Details section contains the following items:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Monitor | DELL 16:9 24inch | 2 | 350.00 | 700..00 | | Dock | USB HDMI DVI Dual Monitor | 1 | 200.00 | 200.00 |   Payment Status shows ‘Paid on <yesterday’s date>  Total Invoice Cost shows ‘$900.00’ | Pass |

## Edit Invoice Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | Edit Invoice | | |
| **Description:** | Tests that an invoice can be edited | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 6.0 | Click on a customer name ‘Jorge’. | Customer name ‘Jorge’ is selected.  Customer details for ‘Jorge’ are displayed:  Customer number (2), company (Yamaha), first name (Jorge), last name (Lorenzo), street address (99 Flinders Ave), suburb (Darwin), state (NT), post code (6061), contact phone (0478221847).  Invoice number 3 is selected.  Invoice Details section contains the following items:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Keyboard | Wireless Keyboard | 3 | 34.99 | 104.97 | | USB Drive | SanDisk 16GB | 4 | 59.00 | 236.00 |   Payment Status shows Overdue. Payment was due on 01/06/2016’  Total Invoice Cost shows ‘$340.97’ | Pass |
| 6.1 | Click on ‘Edit Invoice’ button | The ‘Edit Invoice’ form will be displayed.  ‘Invoice Number’ field is read only and contains 3.  Payment Due Date is set to 01/06/2016.  Payment Status checkbox is unchecked.  Item’s table shows:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Keyboard | Wireless Keyboard | 3 | 34.99 | 104.97 | | USB Drive | SanDisk 16GB | 4 | 59.00 | 236.00 |   Invoice Total shows ‘$340.97’ | Pass |
| 6.2 | Check the Payment Status Paid checkbox.  Set the Date Paid date picker to 07/06/2016  Change the quantity to 1 for the Keyboard item.  Change the quantity to 1 for the USB Drive item. | Payment Status Paid checkbox is checked.  Date Paid date picker shows 07/06/2016  Item’s table shows:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Keyboard | Wireless Keyboard | 1 | 34.99 | 34.99 | | USB Drive | SanDisk 16GB | 1 | 59.00 | 59.00 |   Invoice Total shows ‘$93.99’ | Pass |
| 6.3 | Click the ‘Save’ button | The ‘Edit Invoice’ form disappears.  The main form is displayed.  Customer name ‘Jorge’ is selected in the customer list.  Invoice number 3 is selected in the invoices list.  Invoice Details section contains the following items:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Keyboard | Wireless Keyboard | 1 | 34.99 | 34.99 | | USB Drive | SanDisk 16GB | 1 | 59.00 | 59.00 |   Payment Status shows Paid on 07/06/2016’  Total Invoice Cost shows ‘$93.99’ | Pass |

## Delete Invoice Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | Delete Invoice | | |
| **Description:** | Tests that an invoice can be deleted | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 7.0 | Click on a customer name ‘Jorge’. | Customer name ‘Jorge’ is selected.  Customer details for ‘Jorge’ are displayed:  Customer number (2), company (Yamaha), first name (Jorge), last name (Lorenzo), street address (99 Flinders Ave), suburb (Darwin), state (NT), post code (6061), contact phone (0478221847).  Invoice number 3 is selected.  Invoice Details section contains the following items:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Item | Description | Quantity | Cost | Total Cost | | Keyboard | Wireless Keyboard | 1 | 34.99 | 34.99 | | USB Drive | SanDisk 16GB | 1 | 59.00 | 59.00 |   Payment Status shows Paid on 07/06/2016’  Total Invoice Cost shows ‘$93.99’ | Pass |
| 7.1 | Click on ‘Delete Invoice’ button | A message dialog is displayed showing’ Are you sure you want to delete invoice 3? |  |
| 7.2 | Click ‘Yes’ | The message dialog disappears.  The main form loads.  Customer name ‘Jorge is selected in the customer list.  Invoice number 2 no longer appears in the invoice list.  Invoice Details section contains no items:  Payment Status is empty  Total Invoice Cost is empty | Pass |

## Edit Customer Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | Edit Customer | | |
| **Description:** | Tests that a customer can be edited | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 8.0 | Click on a customer name ‘Marc’. | Customer name ‘Albert’ is selected.  Customer details for ‘Albert’ are displayed:  Customer number (5), company (Microsoft), first name (Albert), last name (Smith), street address (123 Main), suburb (Dianella), state (WA), post code (6000), contact phone (123444). | Pass |
| 8.1 | Click on ‘Edit Customer’ button  Delete ‘Smith’ from last name field.  Type ‘Brown’ in last name field. | The ‘Customer’ form will be displayed.  ‘Customer Number’ field is read only and contains ‘5’.  Last name contains ‘Brown’ | Pass |
| 8.2 | Click the ‘Save’ button | The ‘Customer’ form disappears.  The main form is displayed.  Customer name ‘Albert’ is selected in the customer list.  Last Name now displays ‘Brown’. | Pass |

## Delete Customer Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | Delete Customer | | |
| **Description:** | Tests that a customer can be deleted | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 9.0 | Click on a customer name ‘Dave. | Customer name ‘Dave’ is selected.  Customer number (1), company (Intekka), first name (Dave), last name (Pyle), street address (349 Grand Prom), suburb (Dianella), state (WA), post code (6059), contact phone (0478220117).  Invoice number ‘INV002’ is selected. | Pass |
| 9.1 | Click on ‘Delete Customer’ button | A message dialog is displayed showing’ Are you sure you want to delete customer Dave? | Pass |
| 9.2 | Click ‘Yes’ | The message dialog disappears.  The main form loads.  Customer name ‘Dave’ no longer appears in the customer list.  Customer name ‘Albert’ is selected.  Customer number, company, first name, last name, street address, suburb, state, post code, contact phone for ‘Albert’ are displayed in the Customer Details section. | Pass |

## Search Customers Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | Search by Customer Name | | |
| **Description:** | Tests that a search can be performed on customers by customer name | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 10.0 | Type ‘A’ into the ‘Filter’ field. | All customers apart from those whose names begin with ‘A’ are removed from the customer list.  Customer list shows: Albert, Alex, Andrew, Andy.  Customer name ‘Albert’ is selected.  Customer number, company, first name, last name, street address, suburb, state, post code, contact phone for ‘Albert’ are displayed in the Customer Details section. | Pass |
| 9.1 | Type ‘Al’ into the ‘Filter’ field. | All customers apart from those whose names begin with ‘Al’ are removed from the customer list.  Customer list shows: Albert, Alex  Customer name ‘Albert’ is selected.  Customer number, company, first name, last name, street address, suburb, state, post code, contact phone for ‘Albert’ are displayed in the Customer Details section. | Pass |

## Search Invoices Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | Search by Invoice Number | | |
| **Description:** | Tests that a search can be performed for invoices by invoice number | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 10.0 | Type ‘I’ into the ‘Invoices’ search field. | A pop-up list will appear under the ‘Invoices’ search field showing: INV003, INV004, INV005, INV006, INV007, INV008 | Pass |
| 10.1 | Click on ‘INV003’ in the pop-up list | The ‘Invoices’ search field shows ‘INV003’ | Pass |
| 10.2 | Click the ‘Search’ button | The ‘Invoices’ search field is empty.  ‘Lena’ is selected in the customer list.  Customer number, company, first name, last name, street address, suburb, state, post code, contact phone for ‘Lena’ are displayed in the Customer Details section.  Invoice number ‘INV003’ is selected.  Invoice Details section contains the following items:   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Item Num | Item | Description | Cost | Qty | Total Cost | | 5 | Keyboard | Wireless Keyboard | 34.99 | 3 | 104.97 | | 6 | USB Drive | SanDisk 16Gb | 59.00 | 4 | 236.00 |   Payment Status shows ‘Unpaid. Payment due on <today’s date>’  Total Invoice Cost shows ‘$340.97’ | Pass |
| 10.3 | Type ‘xx’ into the ‘Invoices’ search field.  Click the ‘Search’ button | A message dialog is displayed showing ’No results for invoice number: XX’ | Pass |
| 10.4 | Click ‘OK’ | The message dialog box disappears. | Pass |

## Change Interface Background Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Title:** | Change Theme | | |
| **Description:** | Tests that the interface background image can be changed | | |
|  | | | |
| **ID** | **Instructions** | **Expected** | **Pass/Fail** |
| 10.0 | Select ‘Dark’ from the ‘Theme’ drop down list | Theme field shows ‘Dark’  The background interface of the main form changes to a dark image. | Pass |
| 10.1 | Click the ‘Add Customer’ button | The ‘Customer’ form appears with the same dark background image as the main form. | Pass |
|  | Click the ‘Cancel’ button | The ‘Customer’ form disappears. | Pass |
| 10.1 | Select ‘Light’ from the ‘Theme’ drop down list | Theme field shows ‘Dark’  The background interface of the main form changes to a light image. | Pass |
| 10.2 | Select ‘Default’ from the ‘Theme’ drop down list | Theme field shows ‘Default’  The background interface of the main form changes back to the default background (grey). | Pass |
| 10.3 | Click’ File’ then ‘Exit’. | Application closes. | Pass |

# Annex A – UI Component Details

The tables below detail each component of the UI and a description on the function of the component.

## Login Form UI Components

|  |  |  |
| --- | --- | --- |
| **Component** | **Name** | **Description** |
| Label | lblTitle | Displays ‘ABC Invoice Application’ |
| Group Box | gBoxLogin | Displays ‘Login Details’ group box |
| Label | lblUsername | Displays ‘Username’ |
| Text Box | txtBoxUser | Accepts Username |
| Label | lblPassword | Displays ‘Password’ |
| Text Box | txtBoxPassword | Accepts Password |
| Button | btnLogin | ‘Login’ button |
| Label | lblLoginError | Displays ‘Invalid username or password’ |

## Main Form UI Components

|  |  |  |
| --- | --- | --- |
| **Component** | **Name** | **Description** |
| Menu Strip | msFile | Main menu strip |
| Menu Item | fileToolStripMenuItem | ‘File’ menu item |
| Menu Item | customerToolStripMenuItem | ‘Customer’ menu item |
| Menu Item | addCustomerToolStripMenuItem | ‘Add Customer’ menu item |
| Menu Item | editCustomerToolStripMenuItem | ‘Edit Customer’ menu item |
| Menu Item | deleteCustomerToolStripMenuItem | ‘Delete Customer’ menu item |
| Menu Item | exitToolStripMenuItem | ‘Exit’ menu item |
| Tool Strip | tsMain | Main tool strip |
| Button | btnAddCust | ‘Add Customer’ button |
| Button | btnEditCust | ‘Edit Customer’ button |
| Button | btnDeleteCust | ‘Delete Customer’ button |
| Text Box | tslblTheme | Displays ‘Theme’ |
| Check Box | cboxTheme | Displays/Accepts theme name |
| Label | tsLblInvoices | Displays ‘Invoices’ |
| Text Box | txtBoxInvSearch | Displays/Accepts invoice number search |
| Separator | toolStripSeparator1 | Tool strip separator |
| Separator | toolStripSeparator2 | Tool strip separator |
| Separator | toolStripSeparator3 | Tool strip separator |
| Text Box | gBoxCustomers | Displays ‘Customers’ |
| Label | lblFilter | Displays ‘Filter’ |
| Label | tbSearchCust | Displays/Accepts customer name search |
| Label | lbCustomers | Displays list of customer first names |
| Group Box | gBoxCustomerDetails | Displays ‘Customer Details’ |
| Label | lblCustNum | Displays ‘Customer Number’ |
| Label | lblCompany | Displays ‘Company’ |
| Label | lblFirstName | Displays ‘First Name’ |
| Label | lblLastName | Displays ‘Last Name’ |
| Label | lblStreet | Displays ‘Street Address’ |
| Label | lblSuburb | Displays ‘Suburb’ |
| Label | lblState | Displays ‘State’ |
| Label | lblPostCode | Displays ‘Post Code’ |
| Label | lblPhone | Displays ‘Contact Phone’ |
| Text Box | txtBoxCustNum | Displays customer number |
| Text Box | txtBoxCompany | Displays/Accepts company |
| Text Box | txtBoxFirstName | Displays/Accepts first name |
| Text Box | txtBoxLastName | Displays/Accepts last name |
| Text Box | txtBoxStreet | Displays/Accepts street address |
| Text Box | txtBoxSuburb | Displays/Accepts suburb |
| Text Box | txtBoxState | Displays/Accepts state |
| Text Box | txtBoxPostCode | Displays/Accepts post code |
| Text Box | txtBoxPhone | Displays/Accepts contact phone |
| Group Box | gBoxInvoices | Displays ‘Invoices’ |
| Label | lbInvoiceNum | Displays a list of invoice numbers |
| Button | btnAddInvoice | ‘Add Invoice’ button |
| Button | btnEditInvoice | ‘Edit Invoice’ button |
| Button | btnDeleteInvoice | ‘Delete Invoice’ button |
| Group Box | gbInvoiceDetails | Displays ‘Invoice Details’ |
| DataGridView | dgInvoiceDetails | Invoice details datagridview |
| Label | lblPaymentStatus | Displays ‘Payment Status’ |
| Label | lblStatus | Displays payment status |
| Label | lblTotalInvoiceCost | Displays ‘Total Invoice Cost’ |
| Text Box | tbTotalInvoiceCost | Displays total invoice cost |

## Add Customer, Edit Customer UI Components

|  |  |  |
| --- | --- | --- |
| **Component** | **Name** | **Description** |
| Group Box | gBoxCustFormDetails | Displays ‘Customer Details’ |
| Label | lblCustNumber | Displays ‘Customer Number’ |
| Label | lblCompany | Displays ‘Company’ |
| Label | lblFirstName | Displays ‘First Name’ |
| Label | lblLastName | Displays ‘Last Name’ |
| Label | lblStreet | Displays ‘Street Address’ |
| Label | lblSuburb | Displays ‘Suburb’ |
| Label | lblState | Displays ‘State’ |
| Label | lblPostCode | Displays ‘Post Code’ |
| Label | lblPhone | Displays ‘Contact Phone’ |
| Text Box | txtBoxCustNum | Displays customer number |
| Text Box | txtBoxCompany | Displays/Accepts company |
| Text Box | txtBoxFirstName | Displays/Accepts first name |
| Text Box | txtBoxLastName | Displays/Accepts last name |
| Text Box | txtBoxStreet | Displays/Accepts street address |
| Text Box | txtBoxSuburb | Displays/Accepts suburb |
| Combo Box | cboxState | Displays/Accepts state |
| Text Box | txtBoxPostCode | Displays/Accepts post code |
| Text Box | txtBoxPhone | Displays/Accepts contact phone |
| Label | lblAllReqd | Displays ‘\* all fields required’ |
| Button | btnCustSave | ‘Save’ button |
| Button | btnCancel | ‘Cancel’ button |

## Add Invoice, Edit Invoice UI Components

|  |  |  |
| --- | --- | --- |
| **Component** | **Name** | **Description** |
| Group Box | gbInvoice | Displays ‘Invoice Details’ |
| Label | lblInvoiceNum | Displays ‘Invoice Number’ |
| Label | lblPayDue | Displays ‘Payment Due Date |
| Label | lblDatePaid | Displays ‘Date Paid’ |
| Date Picker | paymentDatePicker | Displays/Accepts payment date |
| Label | lblPayStatus | Displays ‘Payment Status’ |
| DataGridView | dgAddInvoiceItems | Invoice Item datagridview |
| Label | lblInvoiceTotal | Displays ‘Invoice Total’ |
| Text Box | tbInvoiceNum | Displays invoice number |
| Date Picker | paymentDueDatePicker | Displays/Accepts payment due date |
| Check Box | cboxPaidStatus | Displays/Accepts payment status |
| Text Box | txtBoxInvoiceTotal | Displays invoice total |
| Button | btnSave | ‘Save’ button |
| Button | btnCancel | ‘Cancel’ button |

# Annex B – DDL for ABC Invoice Application and Sample Data

## DDL for ANC Invoice Application

DROP SCHEMA IF EXISTS 041110777\_invoice;

CREATE SCHEMA 041110777\_invoice;

USE 041110777\_invoice;

CREATE TABLE Customer (

CustNum smallint(5) unsigned NOT NULL AUTO\_INCREMENT,

FirstName varchar(30) NOT NULL,

LastName varchar(30) NOT NULL,

StreetName varchar(30) NOT NULL,

Suburb varchar(20) NOT NULL,

AddressState varchar(30) NOT NULL,

PostCode varchar(10) NOT NULL,

ContactPhone varchar(30) NOT NULL,

Company varchar(30) NOT NULL,

PRIMARY KEY (CustNum)

);

CREATE TABLE Item (

ItemNum smallint(5) unsigned NOT NULL AUTO\_INCREMENT,

ItemName varchar(30) NOT NULL,

Description varchar(50) NOT NULL,

Cost decimal(10,2) NOT NULL,

PRIMARY KEY (ItemNum)

);

CREATE TABLE Invoice (

InvoiceNum smallint(5) unsigned NOT NULL AUTO\_INCREMENT,

PaymentStatus boolean NOT NULL,

PaymentDate date NOT NULL,

PaymentDueDate date NOT NULL,

CustNum smallint(5) unsigned NOT NULL,

PRIMARY KEY (InvoiceNum),

FOREIGN KEY (CustNum) REFERENCES Customer (CustNum) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE InvoiceItem (

InvoiceNum smallint(5) unsigned NOT NULL,

ItemNum smallint(5) unsigned NOT NULL,

Qty smallint(5) NOT NULL,

PRIMARY KEY (InvoiceNum, ItemNum),

FOREIGN KEY (InvoiceNum) REFERENCES Invoice (InvoiceNum) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (ItemNum) REFERENCES Item (ItemNum) ON DELETE CASCADE ON UPDATE CASCADE

);

DROP PROCEDURE IF EXISTS InsertCust;

DELIMITER //

CREATE PROCEDURE InsertCust(OUT CustNum INT, IN FirstName VARCHAR(30), IN LastName VARCHAR(30), IN StreetName VARCHAR(30), IN Suburb VARCHAR(20), IN AddressState VARCHAR(30), IN PostCode VARCHAR(10), IN ContactPhone VARCHAR(30), IN Company VARCHAR(30))

BEGIN

SET @FirstName = FirstName;

SET @LastName = LastName;

SET @StreetName = StreetName;

SET @Suburb = Suburb;

SET @AddressState = AddressState;

SET @PostCode = PostCode;

SET @ContactPhone = ContactPhone;

SET @Company = Company;

INSERT INTO customer (FirstName, LastName, StreetName, Suburb, AddressState, PostCode, ContactPhone, Company)

VALUES (@FirstName, @LastName, @StreetName, @Suburb, @AddressState, @PostCode, @ContactPhone, @Company);

SET CustNum = LAST\_INSERT\_ID();

END//

DELIMITER ;

DROP PROCEDURE IF EXISTS InsertInv;

DELIMITER //

CREATE PROCEDURE InsertInv(OUT InvoiceNum INT, IN PaymentStatus boolean, IN PaymentDate date, IN PaymentDueDate date, IN CustNum smallint(5))

BEGIN

SET @PaymentStatus = PaymentStatus;

SET @PaymentDate = PaymentDate;

SET @PaymentDueDate = PaymentDueDate;

SET @CustNum = CustNum;

INSERT INTO Invoice (PaymentStatus, PaymentDate, PaymentDueDate, CustNum)

VALUES (@PaymentStatus, @PaymentDate, @PaymentDueDate, @CustNum);

SET InvoiceNum = LAST\_INSERT\_ID();

END//

DELIMITER ;

## Sample Data

INSERT INTO `customer` (`CustNum`, `FirstName`, `LastName`, `StreetName`, `Suburb`, `AddressState`, `PostCode`, `ContactPhone`, `Company`) VALUES

(1, 'Valentino', 'Rossi', '46 Grand Prom', 'Dianella', 'WA', '6059', '0478220117', 'Yamaha'),

(2, 'Jorge', 'Lorenzo', '99 Flinders Ave', 'Darwin', 'NT', '6061', '0478221847', 'Yamaha'),

(3, 'Marc', 'Marquez', '93 Mill Lane', 'Melbourne', 'VIC', '6061', '1512281438', 'Honda'),

(4, 'Dani', 'Pedrosa', '26 Rundle St', 'Adelaide', 'SA', '6000', '0478220114', 'Honda'),

(5, 'Andrea', 'Iannone', '29 Craven Avenue', 'Canberra', 'ACT', '2002', '0456322954', 'Ducati'),

(6, 'Andrea', 'Dovizioso', '4 Edgedale Road', 'Sydney', 'NSW', '6000', '0328475912', 'Ducati'),

(7, 'Bradley', 'Smith', '38 Blossom Avenue', 'Brisbane', 'QLD', '6000', '0858733992', 'Yamaha'),

(8, 'Alex', 'Marquez', '21 Green Ave', 'Fremantle', 'WA', '6800', '0478220118', 'Honda');

INSERT INTO `invoice` (`InvoiceNum`, `PaymentStatus`, `PaymentDate`, `PaymentDueDate`, `CustNum`) VALUES

(1, 1, '2016-01-01', '2016-06-01', 1),

(2, 0, '2016-01-01', '2016-06-01', 1),

(3, 0, '2016-01-01', '2016-06-01', 2),

(4, 1, '2016-05-31', '2016-06-01', 5),

(5, 0, '2016-05-31', '2016-06-04', 6),

(6, 0, '2016-05-31', '2016-06-30', 7),

(7, 0, '2016-05-31', '2016-06-09', 4),

(8, 1, '2016-06-07', '2016-06-11', 5),

(9, 1, '2016-06-06', '2016-06-30', 8);

INSERT INTO `invoiceitem` (`InvoiceNum`, `ItemNum`, `Qty`) VALUES

(1, 1, 2),

(1, 2, 2),

(2, 3, 1),

(2, 4, 2),

(3, 5, 3),

(3, 6, 4),

(4, 1, 2),

(4, 2, 2),

(4, 3, 1),

(5, 2, 3),

(5, 3, 3),

(5, 4, 2),

(6, 2, 1),

(7, 1, 1),

(7, 5, 1),

(8, 2, 1),

(8, 3, 2),

(8, 4, 1),

(9, 1, 1),

(9, 3, 2),

(9, 4, 1);

INSERT INTO `item` (`ItemNum`, `ItemName`, `Description`, `Cost`) VALUES

(1, 'Mouse', 'Wireless Mouse', '14.99'),

(2, 'Macbook Pro', 'i7 8Gb RAM 15-inch', '1450.00'),

(3, 'Monitor', 'DELL 16:9 24inch', '350.00'),

(4, 'Dock', 'USB HDMI DVI Dual Monitor', '200.00'),

(5, 'Keyboard', 'Wireless Keyboard', '34.99'),

(6, 'USB Drive', 'SanDisk 16GB', '59.00');