7 – Developmental Testing

Contents

[Objective Summary 2](#_Toc504376985)

[1: An Objective 3](#_Toc504376986)

[2: An Objective 4](#_Toc504376987)

[3: An Objective 5](#_Toc504376988)

[4: An Objective 6](#_Toc504376989)

# Objective Summary

|  |  |  |  |
| --- | --- | --- | --- |
| # | Objective | Success Criteria | Required Performance |
| 1 | Add Staff | Allow successful addition of staff | Add staff to the program to be used after add staff window closes. |
| 2 | Edit Staff details | Allow staffs info to be edited and changed |  |
| 3 | Delete Staff | Allow staff to be removed from the system | Should have validation window to confirm |
| 4 | View All Staff | Allow a list of staff to be shown | Should take no more than 5 seconds |
| 5 | Save Staff to file | All staffs details will be encrypted and stored to file each time a change is made | Should take no more than 5 seconds |
| 6 | Retrieve Staff from file | Staffs data is able to be decrypted from the file and stored in an array | Should take no more than 5 seconds |
| 7 | Search through Staff | Be able to return a table of results | Search less than 2 seconds for 20 staff |
| 8 | Store the amount of hours staff work | Create / update the store of staff hours |  |
| 9 | Display staff names | Be able to show a list of staff names and details |  |
| 10 | Display staff hours worked per week | show on screen the amount of hours that staff have worked |  |
| 11 | Print staff hours worked per week | Allow printing of staff info ( staff list) |  |
| 12 | Store securely staff logins for the system | Encrypt and decrypt data easily at certain points of the program |  |
| 13 | Encrypt passwords | To store the data securely to help prevent changes being made when the staff don’t have the access rights to be able to do so |  |
| 14 | Decrypt passwords before used | Successfully and accurately decrypt passwords |  |
| 15 | Allow different user access rights | allows different access to the program depending upon your role |  |
| 16 | Filter by Name | Allows the data to be sorted and showed in a table |  |
| 17 | Filter by hours worked | Allows the data to be sorted and showed in a table |  |
| 18 | Sort by Name | Data is sorted by name Ascending or Descending and then stored permanently that way |  |
| 19 | Allow different views of data | Views of data change depending upon role of staff that is logged in |  |
| 20 | Backup staff details | Save the staff details in a file that is encrypted and also is stored away from the program |  |
| 21 | Access Level (view staff info) - Admin | Allow only admins to access certain sets of data |  |
| 22 | Access Level (view staff names) - normal | Allow any staff to view certain sets of data |  |
| 23 | Access Level (change staff info) - Admin | Allow only Admins to change certain sets of data |  |
| 24 | Reject incorrect dates – (eg. 31st feb) | If the date is wrong then the program shouldn’t accept it |  |
| 25 | Separate views for different access levels – normal only view names etc  Admin can view all and edit | Allow users different views of data, depending upon what role they are. |  |
| # | Objective | Success Criteria | Required Performance |
| 26 | Add Customer | Allows customers to be added to the program | Instantly added and be able to be used |
| 27 | Edit Customer Details | Allows the data stored in the program to be edited with the correct access rights |  |
| 28 | Delete Customers | Allows customers to be deletes with the correct access rights |  |
| 29 | View all Customer | Allows a scrollable list of customers to be shown on the screen |  |
| 30 | Encrypt sensitive information | Sensitive info is saved only when it is encrypted and is decrypted just before it is used | Encryption takes no more than 1 second per 5 items |
| 31 | Decrypt Encrypted information | Sensitive info can be decrypted before it is used | Decryption takes no more than 1 second per 5 items |
| 32 | Sort customers | List of customers can be sorted by name and saved permanently to file / array | Should take no more than 5 seconds |
| 33 | Search customers | Allows the list of customers to be searched through and searched by name etc | Search should take no more than 5 seconds |
| 34 | Store complaints | Able to store the complaints of the customers and create a log of it |  |
| 35 | Store returns | Store the returns in a file and be able to retrieve it |  |
| 36 | Store customers securely | Encrypt the data stored on customers and save it to a file |  |
| 37 | Backup customers details | Able to store customers data in a separate location to the program |  |
| 38 | Output list of customers | Print or show on screen a list of customers and their data, depending upon what is selected ( also based on access level) |  |
| 39 | Display lists of customers | Show on screen a list of customers in a table on the screen – data is dependent on access level of the user |  |
| 40 | Search by name | Search for a customer by their first name | Return a result within 3 seconds |
| 41 | Search by Last Name | Search for a customer by their last name | Return a result within 3 seconds |
| 42 | Access Level - normal | Allow viewing and editing of data with normal access level. |  |
| 43 | Reject fake info – dates that don’t exist (eg 30th Feb) | Don’t allow data that is of wrong format or is clearly incorrect be accepted into the database | Be able to identify dates that don’t exist within the program |
| # | Objective | Success Criteria | Required Performance |
| 44 | Add return | Returns can be added to the program | Data added should be able to be accessed right away |
| 45 | Edit return | Able to have data edited |  |
| 46 | Delete return | Data should be able to be deleted |  |
| 47 | Encrypt return data | Data should be stored securely | Should happen within 2 seconds |
| 48 | Store returns | Saved to a file |  |
| 49 | Read returns from file | Get the returns from the file and read into the program |  |
| 50 | Decrypt returns while reading into system | Able to decrypt before using data |  |
| 51 | View all returns | Show all returns on screen | Retrieve within 2 seconds |
| 52 | Search by product | Search products |  |
| 53 | Print Return list | Print off list of returns |  |
| 54 | Display all returns | Show all returns |  |
| 55 | Search by return date | Search the list of returns by date and return |  |
| 56 | Sort by product | Sort the list by product |  |
| 57 | Sort by date of return | Sort the list by date of return |  |
| 58 | Access level - normal |  |  |
| 59 | Output return info | Show the info of the returns on the screen |  |
| 60 | Generate return ID | Make a unique return id |  |
| 61 | Reject return that is out of warranty (unless admin override) | Not allow products out of warrenty to be returned |  |
| # | Objective | Success Criteria | Required Performance |
| 62 | Add clock times | Add clock times to the system |  |
| 63 | Edit clock times | Admin only edit clock times |  |
| 64 | Delete clock times | Delete the clock times from the list ( admin only) |  |
| 65 | Encrypt clock data data | Encrypt data whenever it isn’t used |  |
| 66 | Store clock data | Store the encrypted data |  |
| 67 | Read clock data from file | Read the clock data from the file |  |
| 68 | Decrypt clock data while reading into system | Decrypt the clock data before using it |  |
| 69 | Calculate hours worked | Calculate the total hours worked per week for each employee |  |
| 70 | Access Level - Admin | Only admins can access this data |  |
| 71 | Display hours worked | Show total hours worked |  |
| 72 | Interface easy to understand / use | The interface will be easy to use |  |
| 72 | Check that staff have clocked in before clocking out | Wont allow clock out before clock in |  |
| 74 | Reject clock out without clock in | Wont allow clock out without clock in |  |
| 75 | Reject clock in after 5:00PM | Wont allow clock in after 5:00PM |  |

# 1: Add Staff

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Stage | Result / Problem | Evidence | Action and Justification  (If Needed) | Evidence of Action  (If Needed) |
| 0 | Soft Dev  /  Proto | Username came up as stars instead of displaying text |  | Changed the username component to a JTextField from a JPasswordField |  |
| 1 | *Prototype* | *Error ; expected* |  | I added the ; where it was expecting it |  |
| 2 | *Prototype* | *Error missing return statement* |  | I added a return statement to the method |  |
| 3 | *Prototype* | jButtons showing up when they shouldn’t |  | I had added the tabbed pane that has the buttons on it to the frame as well as the panel on top of it so it wasn’t working properly. So oi have removed the line of code |  |
| 4 | *Prototype* | Stopped the tabbed pane from loading at all. | -No Picture- | To fix this I re added the line and added a blank tab to the tabbed pane and set that to be the one that it is set to when the program starts. This blank pane is what it is, it is a completely blank page, so nothing can show up that isn’t supposed to |  |

# 2: Edit Staff Details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Stage | Result / Problem | Evidence | Action and Justification  (If Needed) | Evidence of Action  (If Needed) |
| 1 | Soft Dev  /  Proto |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |

# 3: Delete Staff

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Stage | Result / Problem | Evidence | Action and Justification  (If Needed) | Evidence of Action  (If Needed) |
| 1 | Soft Dev  /  Proto | I didn’t initialise the temporary array correctly, and this is causing some errors |  | I have initialised the array correctly |  |
| 2 | *Proto* | *I need to convert integers to a string before I try to use them as a parameter* |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |

# 4: view all staff

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Stage | Result / Problem | Evidence | Action and Justification  (If Needed) | Evidence of Action  (If Needed) |
| 1 | Soft Dev  /  Proto | Username came up as stars instead of displaying text |  | Changed the username component to a JTextField from a JPasswordField |  |
| 2 |  | *I made and a panel to view the staff in the program.* |  | Everything was squashed to the side even though I had set them to be more spaced horizontally…  I realised that I had missed a crucial component: viewStaff.setLayout(null); so I added it and now it works |  |
| 3 | *Proto* | I tested the table and button for a new panel – viewStaff  And another test for the layout |  | It worked |  |
| 4 |  | *next test for same objective* |  |  |  |

# 5. Save staff to file

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Stage | Result / Problem | Evidence | Action and Justification  (If Needed) | Evidence of Action  (If Needed) |
| 1 | Prototype | I misspelled the variable name for the filename |  | Changed the variable name for the filename in the |  |
| 2 | *Prototype* | *Success – this is the saved file* |  |  |  |

# 6. Retrieve staff from file

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Stage | Result / Problem | Evidence | Action and Justification  (If Needed) | Evidence of Action  (If Needed) |
| 1 | Prototype | I misspelled the variable name for the filename |  | Changed the variable name for the filename in the |  |
| 2 | *prototype* | *success* |  |  |  |

# 7.Search through staff

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Stage | Result / Problem | Evidence | Action and Justification  (If Needed) | Evidence of Action  (If Needed) |
| 1 | Prototype | Misspelt the name of the variable for the result of the search within the method to search for staff |  | I changed the code so that the correct variable names are used |  |
| 2 | *Prototype* | *I misspelt the variable name for the result of the search and also the next search location* |  | I looked over the names and changed the name to be resultOfStaffSearch and nextSearchLocation |  |
| 3 | *Prototype* | *The compiler cannot find the symbol for the search parameter and the . for integer.parseInt* |  | I changed the variable names and forgot to use camelCase on one occasion, I have changed the parameter in the method to be camel case now | N/A |
| 4 | *Prototype* | *The compiler didn’t recognise the parseint method.* |  | I had misspelt the parseInt to be ParseInt, this is why it wasn’t getting recognised |  |

# 8. Store the amount of hours’ staff work

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Stage | Result / Problem | Evidence | Action and Justification  (If Needed) | Evidence of Action  (If Needed) |
| 1 | Soft Dev  /  Proto |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |

# 9. Display staff names

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Stage | Result / Problem | Evidence | Action and Justification  (If Needed) | Evidence of Action  (If Needed) |
| 1 | Soft Dev  /  Proto |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |

# 10. Display staff hours worked per week

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Stage | Result / Problem | Evidence | Action and Justification  (If Needed) | Evidence of Action  (If Needed) |
| 1 | Soft Dev  /  Proto |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |

# 62. Add clock times

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | Stage | Result / Problem | Evidence | Action and Justification  (If Needed) | Evidence of Action  (If Needed) |
| 1 | Proto | The switch that I tried to implement needs the conditions to be known to be able to compile it, the items that I tried to switch between were not pre-defined. They would be worked out after the program starts, so therefore this wouldn’t work. |  | I changed the switch that I wanted to use for a simple if statement, as if statements don’t require the outcomes to be already calculated during compilation. |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |