|  |  |  |
| --- | --- | --- |
| Icon  Description automatically generated | **CMPS 350 Project Phase 2 – UI Design Implementation (10% of the course grade)** | |
| **Group Id:** | | G7 |
| **Group Members:** | | Hajar Tariq Alyafei  Amna Abdalla Sheikh Mohamed Mohamed  Noora Bakhit Al-Marri  **Emails:** [ha1802322@student.qu.edu.qa](mailto:ha1802322@student.qu.edu.qa)  [am1803081@qu.edu.qa](mailto:am1803081@qu.edu.qa)  [200759398@qu.edu.qa](mailto:200759398@qu.edu.qa) |

**Grading Rubric - In the Functionality column please specify either: *Working (completed x%)*, *Not Working (completed x%)* or *Not done*.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Criteria | | % | Functionality\* | Quality of the implementation | Your Grade |
| **1)** Design and implementation of the database schema to manage the data in a MongoDB database (including a schema diagram in the project report). | | 10% | 10% |  |  |
| **2)** Populate the database with the data from the json files. | | 5% | 5% |  |  |
| **3) Implementation:**  - Repository Implementation to read/write data from/to MongoDB (30%)  - Web API implementation (25%)  - Change Web UI to use the Web API (20%) | | **75%** | 75% |  |  |
|  | U1 – Login | 5 | 5% |  |  |
| U2 - YalaPay Dashboard | 10 | 10% |  |  |
| U3 - List/Search/Add/ Update and Delete a Customer:  1. List Customers (5 pts)  2. Search Customers (5 pts)  3. Add Customer (5 pts)  4. Update Customer (5 pts)  5. Delete Customer (5 pts) | 25 | 25% |  |  |
| U4 - List/Search/Add/ Update and Delete an Invoice  1. List Invoices (5 pts)  2. Search Invoices (5 pts)  3. Add Invoice (5 pts)  4. Update Invoice (5 pts)  5. Delete Invoice (5 pts) | 25 | 25% |  |  |
| U5 - List/Search/Add/ Update and Delete payments for an invoice  1. List Payments (5 pts)  2. Search Payments (5 pts)  3. Add Payment (5 pts)  4. Update Payment (5 pts)  5. Delete Payment (5 pts) | 25 | 25% |  |  |
| U6 - Manage Cashing Cheques  1. Add Cheques Deposit (10 pts)  2. List Cheque Deposits (5 pts)  3. Update Cheques Deposit (15 pts) | 30 | 30% |  |  |
| U7 - Invoices Report | 10 | 9.75% |  |  |
| U8 - Cheques Report | 10 | 9.75% |  |  |
| **4) Design and Testing Documentation**  **\* Design documentation:**  - Web API services diagram  - Database schema diagram  **\* Testing documentation:** with evidence of working implementation using snapshots illustrating the results of your solution testing.  \* **Discussion of the project contribution** of each team member [-10pts if not done] | | 10% | 10% |  |  |
| **Total** | | 100 |  |  |  |
| Copying and/or plagiarism or not being able to explain or answer questions about the implementation | | -  100 |  |  |  |

**\* Possible grading for functionality** - ***Working*** (get 70% of the assigned grade), ***Not*** ***working*** (lose 40% of assigned grade and ***Not done*** (get 0). The remaining grade is assigned to the quality of the implementation.

In case your implementation is not working then 40% of the grade will be lost and the remaining 60% will be determined based on of the code quality and how close your solution to the working implementation.

Solution quality also includes meaningful naming of identifiers (according to Android naming conventions), no redundant code, simple and efficient design, clean implementation without unnecessary files/code, use of comments where necessary, proper code formatting and indentation.

**Marks will be reduced** forcode duplication, poor/inefficient coding practices, poor naming of identifiers, unclean/untidy submission, and unnecessary complex/poor user interface design.

# Application Design

# Web API services diagram

|  |  |  |
| --- | --- | --- |
| **HTTP Verb** | **Url** | **Functionality** |
| Get | /api/login/:email/:password | Return a specific username depending in email |
| Post | /api/login/:email/:password | Add username |
| Get | /api/customers | Returns all customers |
| Put | /api/customers | Update a customer |
| Get | /api/customer/name/:companyName | Return an customer with same company name |
| Get | /api/customer/:customerId | Return customer by customerId |
| Delete | /api/customer/:customerId | Deletes an account by customerId |
| Post | /api/customers | Adds a customer |
| Get | /api/invoices | Return all invoices |
| Post | /api/invoices | Add an invoice |
| Put | /api/invoices | Update an invoice |
| Get | /api/invoice /:InvoiceNo | Return invoice by InvoiceNo |
| Delete | /api/invoice/:InvoiceNo | Deletes an invoice by InvoiceNo |
| Get | /api/invoices/sum | Return the total of all invoices |
| Get | /api/invoices/sum/:startDate/:endDate | Return the total of invoices with same start Date and end Date |
| Get | /api/invoices/sum/:afterMonth | Return the total of invoices that complete 30 day |
| Get | /api/invoices/search/: status /:toDate/:fromDate | Return all invoices depending in status of invoice and specific date |
| Get | /api/invoice/balance/:InvoiceNo | Return balance for invoice which have same InvoiceNo |
| Get | /api/invoices/search/: toDate/:fromDate | Return all invoices depending in specific date |
| Get | /api/payments | Return all payments |
| Post | /api/payments | Add a payment |
| Put | /api/payments | Update a payment |
| Get | /api/payment/:paymentId | Return payment by paymentId |
| Delete | /api/payment/:paymentId | Delete payment by paymentId |
| Get | /api/payments/:invoice/:invoiceNo | Get payments with same invoice number |
| Get | /api/deposits | Return all deposits |
| Post | /api/deposits | Add a deposit |
| Put | /api/deposits | Update a deposit |
| Get | /api/deposit/:depositId | Return deposit by depositId |
| Delete | /api/deposit/:depositId | Delete deposit by depositId |
| Get | /api/cheques | Return all cheques |
| Post | /api/cheques | Add a cheques |
| Get | /api/cheque/:ChequeNo | Return cheque by ChequeNo |
| Put | /api/cheque/:ChequeNo | Update cheque with same ChequeNo |
| Delete | /api/cheque/:ChequeNo | Delete cheque with same ChequeNo |
| Get | /api/cheques/sum | Return total of all cheques |
| Get | /api/cheques/sum/:status | Return total of all cheques depending in status of cheques |
| Get | /api/cheques/:status /:toDate/:fromDate | Return all cheques with specific status and specific specific dates |
| Get | /api/cheques /:toDate/:fromDate | Return all cheques from specific dates |
| Get | /api/banks | Return all banks |
| Post | /api/banks | Add a bank |
| Get | /api/BankAccounts | Return all Bank Accounts |
| Post | /api/BankAccounts | Add a Bank Accounts |
| Get | /api/paymentModes | Return all payments mode |
| Post | /api/paymentModes | Add a payment mode |
| Get | /api/returnReasons | Return all return Reasons |
| Post | /api/returnReasons | Add a return Reasons |

# Database schema diagram

Diagram

Description automatically generated

# Testing

# U1 – Login

Correct credential

Graphical user interface, application

Description automatically generated

Test wrong credential

Graphical user interface, application

Description automatically generated

# U2 - YalaPay Dashboard

Graphical user interface

Description automatically generated

# U3 - List/Search/Add/ Update and Delete a Customer

# List Customers

Graphical user interface

Description automatically generated with low confidence

# Search Customers

Graphical user interface, text, application, email

Description automatically generated

# Add Customer

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

# Update Customer

Retrieve data automatically

Graphical user interface, application

Description automatically generated

Updated new customer email



Graphical user interface, application

Description automatically generated

Check if the data updated in db

A screenshot of a computer

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

# Delete Customer

Graphical user interface

Description automatically generated with medium confidence

Delete from database

Graphical user interface, text, application, email

Description automatically generated



# U4 - List/Search/Add/ Update and Delete an Invoice

# List Invoices

Graphical user interface, application

Description automatically generated

# Search Invoices

Graphical user interface, text, application, email

Description automatically generated

# Add Invoice

Graphical user interface, application

Description automatically generated

# Update Invoice

Update invoice 4 from 90000 to 40000 amount

Graphical user interface, application

Description automatically generated

# Delete Invoice

Delete invoice 4

Graphical user interface, application

Description automatically generated

# U5 - List/Search/Add/ Update and Delete payments for an invoice

# List Payments

Graphical user interface, application

Description automatically generated

# Search Payments

Graphical user interface, text, application, email

Description automatically generated

# Add Payment

Graphical user interface

Description automatically generated

# Update Payment

Update paymentID 6 from bank to credit card and change amount from 300 to 400

Graphical user interface

Description automatically generated

# Delete Payment

Delete paymentId 6

Graphical user interface, application

Description automatically generated

# U6 - Manage Cashing Cheques

# Add Cheques Deposit

ADD DepositId 5

Table

Description automatically generated with medium confidence

# List Cheque Deposits

Table

Description automatically generated

# Update Cheques Deposit

Update DepositId 5

Table

Description automatically generated

# U7 - Invoices Report

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

# U8 - Cheques Report

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, Word, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

# Summary of team member contributions

we had a good experience working on this project which is taught us a lot of new things as an aspiring web developer. Our total experience was incredibly intriguing and unique, as we had never previously created web pages from scratch.

Compared to phase one, phase two was more easier and took less time to solve. In this phase, we learned how to fetch data from different datatype such as String, images, Date, and So on from JSON files using MongoDB which is define schemas to model documents. Then use the model to read or write documents and do some CRUD Operations. These Operations are adding new data, updating stored data, deleting from DB., and searching for specific data. Also, do some simple mathematics operation such as add and subtract, to calculate the amount or total sum of some data and show it in the dashboard as statistical data .

we are satisfied with our team; we were all diligent students who were always glad to help each other’s, and we split the tasks among ourselves fairly and effectively. Sometimes, we meet on “teams” to solve some parts together and explain to each other that sometimes was helpful to finish our tasks in less time.

One of the most challenging problems we've had in phase2 is working on another team project, which takes from us some time to find some functions or variables, sometimes understand what the output of a function - study some function- since they use a different way from our work. On other hand, we learned another way to solve phase one and saw a different solution for the project which helped us to develop our critical thinking.