**Web Applications Development – Summer Retake Assignment.**

**Project Definition:**

This application is to be used by the water utility to enable subscribers submit their monthly water readings and receive their monthly water bills. Once the subscriber has submitted their meter reading, the system shall compute the amount the subscriber is to pay and present them with a bill. The subscriber may then elect to pay their bill. A subscriber is able to see the total amount they owe the water utility and the status of all their previous water bills.

**Use Cases:**

1. **Login** – This use case enables a subscriber to log into the application. The subscriber uses a combination of their first name and last name. (Subscriber John Doe has username jdoe) and their meter number as the password.
2. **Capture Meter Reading** – This use case enables a subscriber to capture their meter reading. The subscriber is provided with a form to enter their current meter reading and submit it.
3. **View** **Water Bills** – This use case shows the subscriber a list of all their bills. A bill is created using the previous meter reading, most recent reading, and the units are computed and the amount due calculated. The bill has a date similar to that on which the reading was made.

**Task:**

1. You are to develop a web application composed on a back end that is made up a MySQL database and a RESTFul API either in PHP, Python or NodeJS and a front end with HTML, CSS, JS. (You may use JavaScript frameworks such as React, Bootstrap etc.)
2. You will then build a front end and consume these API endpoints to be able to implement the application defined above

|  |  |  |  |
| --- | --- | --- | --- |
| **Definition** | **Endpoint** | **Request Payload** | **Response** |
| Login – This endpoint enables the user to log into their account |  |  |  |
| User Details – This endpoint provides the user account details for the user dashboard including the full names of the user, their meter number, location of property and their total outstanding balance |  |  |  |
| Submit Readings – This endpoint allows a user to submit their meter readings |  |  |  |
| View Bills – This endpoint returns a list of the user’s bills |  |  |  |
| View Bill – This endpoint returns a particular bill to the customer showing all the details of the bill |  |  |  |

**Deliverables**

1. All your application code in a repository on github or gitlab. You will submit a link to the repository
2. A link to your project hosted on heroku.
3. Documentation of your API showing the endpoints and the request payload and expected response. (Complete the table above)

**Guides**

|  |  |
| --- | --- |
| **Topic** | **Resources** |
| RESTFul API Development with Python | <https://www.roytuts.com/python-rest-api-crud-example-using-flask-and-mysql/> |
| RESTFul API Development with Python | <https://www.nintyzeros.com/2019/11/flask-mysql-crud-restful-api.html> |
| RESTFul API Development with PHP | <https://www.webslesson.info/2018/05/how-to-make-simple-crud-rest-api-in-php-with-mysql.html> |
| RESTFul API Development with PHP | <https://www.w3jar.com/crud-rest-api-in-php-pdo/> |
| Hosting a PHP & MySQL Application on Heroku | <https://scotch.io/@phalconVee/deploying-a-php-and-mysql-web-app-with-heroku> |
| Hosting a Python & MySQL Application on Heroku | <https://www.roytuts.com/how-to-deploy-python-flask-mysql-based-application-in-heroku-cloud/> |

**Use Case Documentation**

Use Case: Login

Participating Actors

The use case is initiated by a Subscriber.

Brief Description

The use case allows the subscriber to log into the MyWaterBill App by providing a username and password.

Flow of Events

Basic Flow

1. The use case starts when the **subscriber** accesses the system using its home page and elects to log in
2. The system asks the subscriber to provide a username and password.
3. The subscriber provides a username and password.
4. The system validates the username and password to ensure that they match the company’s records.
5. The system welcomes the subscriber and presents the subscriber’s dashboard that contains all their information in card and a list of their last 5 bills.
6. After verifying that he has the right information in his account, the subscriber elects to log out.
7. The system returns to its home page. The use case ends.

Alternative Flows

* A1 - Login. In step 4, if the subscriber provides a username and password that do not match the records, the system rejects them and returns to step 2.

Rules

* **Username Rule**: Usernames are provided by users and are a combination of the first letter of the subscriber’s firstname and their entire second name.
* **Password Rule**: Passwords are provided by users and should match the subscriber’s meter number.
* **Basic Security:** authentication should follow standard security guidelines and practices. Passwords should not be sent in the clear, and must be encrypted. It’s up to you how to handle authentication and reconcile it with the RESTful API.

Use Case: Capture Meter Reading

Participating Actors

The use case is initiated by a Subscriber.

Brief Description

The use case allows the subscriber to capture their latest water meter reading into the MyWaterBill App. This use case assumes that the subscriber is already logged in.

Flow of Events

Basic Flow

1. The use case starts when the **subscriber** is o their dashboard inside the app and elects to enter new reading.
2. The system asks the subscriber to enter their current meter reading.
3. The subscriber provides meter reading and submits.
4. The system validates the entered metering reading and asks the subscriber to confirm that it is their current reading.
5. The subscriber confirms that is it the correct meter reading.
6. The system presents the user with a thank you message
7. The system redirects the user to their bills list with the most current bill on top of the list.
8. The user may elect to click on the top most bill to view its details. The use case ends here.

Alternative Flows

* A1 - In step 3, if the subscriber provides a meter reading that is less than the last meter reading in the system, the system rejects it and asks the subscriber for a correct reading.

Rules

* **Bill Rule**: Once the subscriber has submitted the correct meter reading, the system computes the cubic meters of water used by the subscriber and presents them with a bill of the total amount due. The bill is computed using the following rules.
  + **Qty Used;** Latest Reading – Last Reading
  + **Pricing Per Cubic Meter:**

|  |  |
| --- | --- |
| **Consumption** | **Price/M3 in RWF** |
| First 5m3 | 340 |
| Next 5-20m3 | 720 |
| Next 20 – 50m3 | 845 |
| >50m3 | 877 |

* Bill Details – The bill contains the following details;
  + **Bill Date:** The date on which subscriber provides meter reading and submits.
  + **Previous Reading –** last reading before most current reading
  + **Current Reading –** most current reading
  + **Consumption –** M3 of water used
  + **Cost –** The cost of the water.