**Requirement Specification**

**Title Requirements Specification (RS)**

Document Control

Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Scope of Activity** | **Prepared** | **Reviewed** | **Approved** |
| 24/02/15 | 1 | Create | ET | X | X |
| 07/03/2015 | 5 | Use Case Scenarios | MR | X | X |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Version** |
| Niall Farren |  |  |
| Emmett Traynor |  |  |
| Mark Ryan |  |  |
| Daniel Collins |  |  |
|  |  |  |

Related Documents

|  |  |
| --- | --- |
| **Title** | **Comments** |
| Title of Use Case Model |  |
| Title of Use Case Description |  |

**Table of Contents**

[**1 Introduction**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.tyjcwt)

[**1.1 Purpose**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.3dy6vkm)

[**1.2 Project Scope**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.1t3h5sf)

[**1.3 Definitions, Acronyms, and Abbreviations**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.4d34og8)

[**2 User requirements definition**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.2s8eyo1)

[**3 System architecture**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.17dp8vu)

[**4 Requirements specification**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.3rdcrjn)

[**4.1 Physical environment requirements**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.26in1rg)

[4.1.1 Requirement 1 <name of requirement in a few words>](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.lnxbz9)

[4.1.2 Description & Priority](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.35nkun2)

[**4.2 Interface requirements**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.1ksv4uv)

[4.2.1 Requirement 1 <name of interface requirement in a few words>](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.44sinio)

[4.2.2 Description & Priority](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.2jxsxqh)

[**4.3 Functional requirements**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.z337ya)

[4.3.1 Requirement 1 <name of requirement in a few words>](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.3j2qqm3)

[4.3.2 Requirement 1 <name of requirement in a few words>](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.1y810tw)

[**4.4 Documentation requirements**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.4i7ojhp)

[4.4.1 Requirement 1 <name of document requirement in a few words>](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.2xcytpi)

[4.4.2 Description & Priority](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.1ci93xb)

[**4.5 Data requirements**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.3whwml4)

[4.5.1 Requirement 1 <name of data requirement in a few words>](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.2bn6wsx)

[4.5.2 Description & Priority](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.qsh70q)

[**4.6 Non-Functional Requirements**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.3as4poj)

[4.6.1 Performance/Response time requirement](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.1pxezwc)

[4.6.2 Availability requirement](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.49x2ik5)

[4.6.3 Recover requirement](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.2p2csry)

[4.6.4 Robustness requirement](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.147n2zr)

[4.6.5 Security requirement](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.3o7alnk)

[4.6.6 Reliability requirement](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.23ckvvd)

[4.6.7 Maintainability requirement](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.ihv636)

[4.6.8 Portability requirement](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.32hioqz)

[4.6.9 Extendibility requirement](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.1hmsyys)

[4.6.10 Reusability requirement](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.41mghml)

[4.6.11 Resource utilization requirement](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.2grqrue)

[**5 System models**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.vx1227)

[**6 System evolution**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.3fwokq0)

[**7 Appendices**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.1v1yuxt)

[**7.1 Use case 1**](https://docs.google.com/document/d/1inD8wvtWfj5gqyydqEQ0PL0U8CJaY6tRgMkd0M--ykI/edit#heading=h.4f1mdlm)

## Introduction

* 1. **Purpose**

The purpose of this document is to set out the requirements for the development of a Group Water Scheme Application (GWS App) which will provide a billing and payment service for the groups committee and members respectively. The intended customers are primarily the administration staff of the Billis/Lavey Group Water Scheme in Cavan and its 600+ domestic users.

* 1. **Project Scope**

The scope of the project is to develop an online application which shall provide a billing platform for domestic water meter’s and the subsequent payment of said bills.

One of our team members is involved in the annual billing administration of said group water scheme and is aware of the requirements needed in order to automate the existing process.

The application will provide each member with an account within which a breakdown of each year’s water fees can be seen. Members will be provided with a unique username and password. A facility will exist for members to top up their accounts in order to keep a credit balance. From a back-end perspective, it will be possible for administrators to; add new client records, amend existing client records, print single records, print multiple records, email account statements to clients, edit rates and add new rate bands.

There is a time constraint on this project insofar as it needs to be complete circa end of April 2015 in order to meet the project submission deadline. There are several features we may wish to add to the software functionality in time, but will fall outside the project scope.  We will note these where applicable.

The application will be hosted in the cloud and the design should be responsive so that it will render appropriately on mobile and handheld devices.

* 1. **Definitions, Acronyms, and Abbreviations**

GWS Group Water Scheme  
SaaS Software as a Service  
User Someone who interacts with the application  
Admin Someone who has specific permission for managing and controlling the system

**2. User requirements definition**

Requirements for a web application specify what we want or desire from the application. Ultimately the implementation of the application should lead to a business advantage.  This advantage need not be just a cost reduction; it could be a reduction in the time taken to process something. In this instance it is both. The following section describes the set of objectives and requirements for the system from both the administrator and user perspectives.

**3. System architecture**

The software architecture will be built in adherence with an MVC/MVP structure using C# and the Microsoft .NET framework. Data persistence will be provided by an SQL database via the Entity Framework relational database management system.

The application will conform to a client-server model. The administrator of the system and other users will log in externally using a thin client (standard web browser) which will not require extensive local resources.

In terms of hardware, the application is intended to be deployed to a third party cloud provider (Microsoft Azure if possible), so no web server or networking hardware will be required locally. Potentially it could alternatively be hosted by the administrator in the traditional manner if preferred, using IIS or another server technology.

**4. Requirements specification**

All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

The following table demonstrates how to make requirements verifiable incorporate metrics as follows:

* 1. **Physical environment requirements**

The application is intended to be hosted by an external cloud provider, so physical environment requirements will be handled by the third party host.

* 1. **Interface requirements**

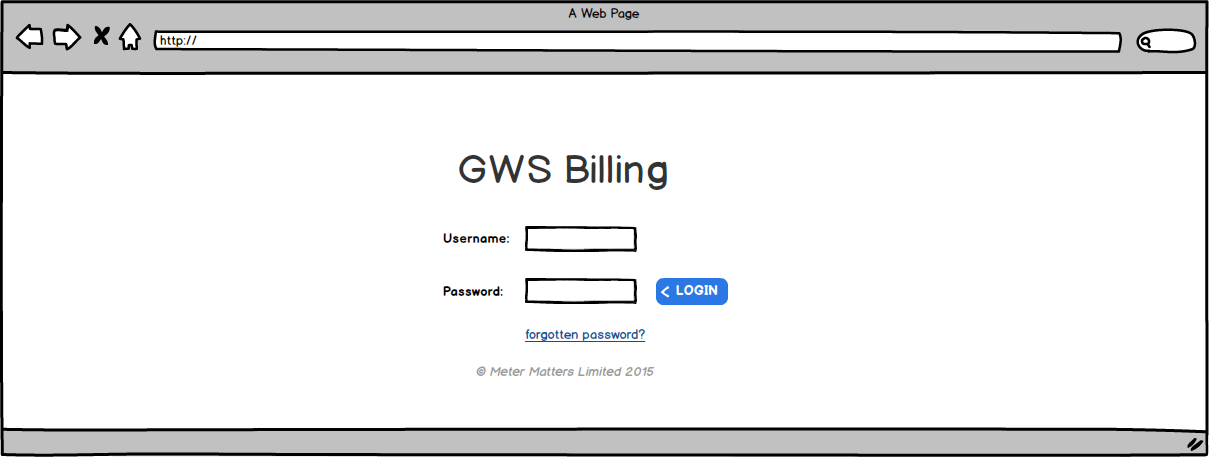
The application should be capable of exporting (and possibly importing) data in csv and possibly other formats (excel etc). PDF functionality is also required for exporting printable invoices/statements. Email functionality will be included through the SMTP protocol. As a web application it will implement full internet functionality and will be accessible on any standard internet-capable device via HTTP. The database interface will be handled by the .Net Entity Framework ORM.

**Requirement 1: Graphical User Interfaces - Login**

**Description & Priority**

A first-time user of the application should see the log-in page when they browse to the URL. All users will be pre-registered and will receive login credentials from the Group Water Scheme via post. Users will be requested to change their passwords upon their first login. This is a high-priority requirement.

Mock-Up 1:

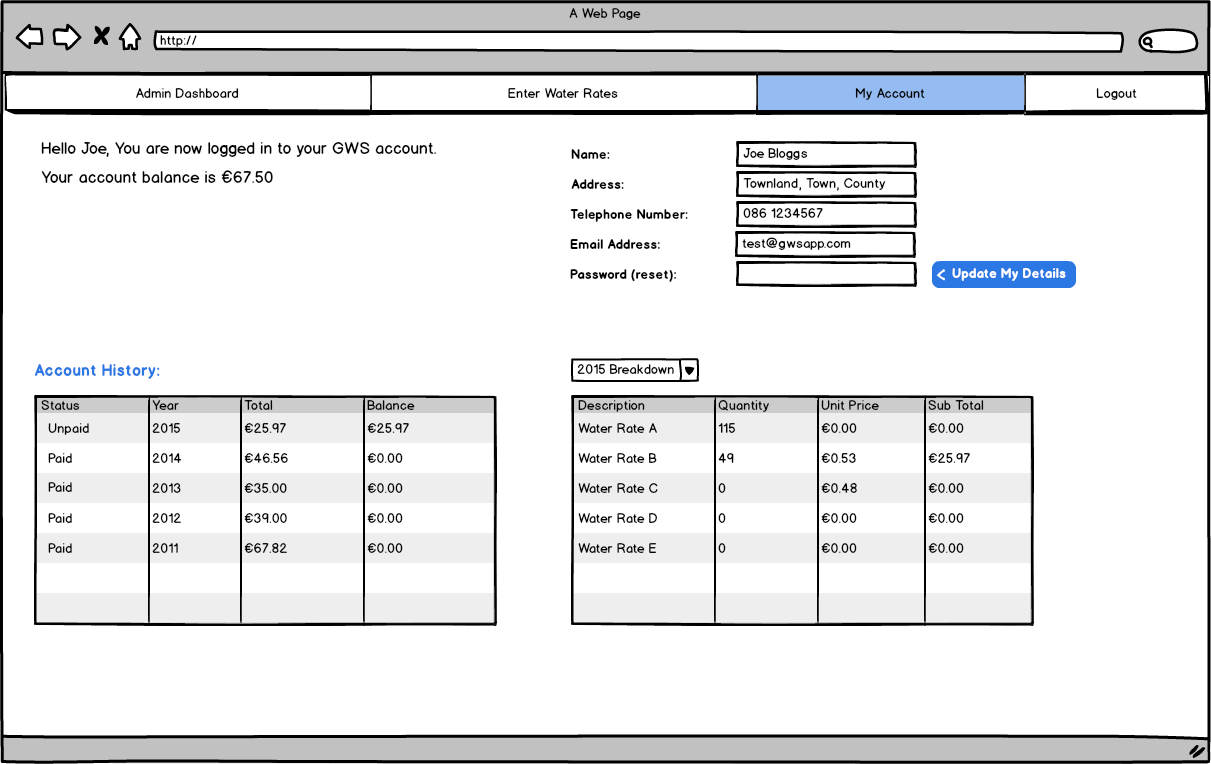
****

**Requirement 2: Graphical User Interfaces - MyAccount**

**Description & Priority**

Every user should have a profile page where they can update their name, address, e-mail address, phone number and password. The ‘MyAccount’ page will also display details of the users account history for the required year. Displaying of account information is high-priority.

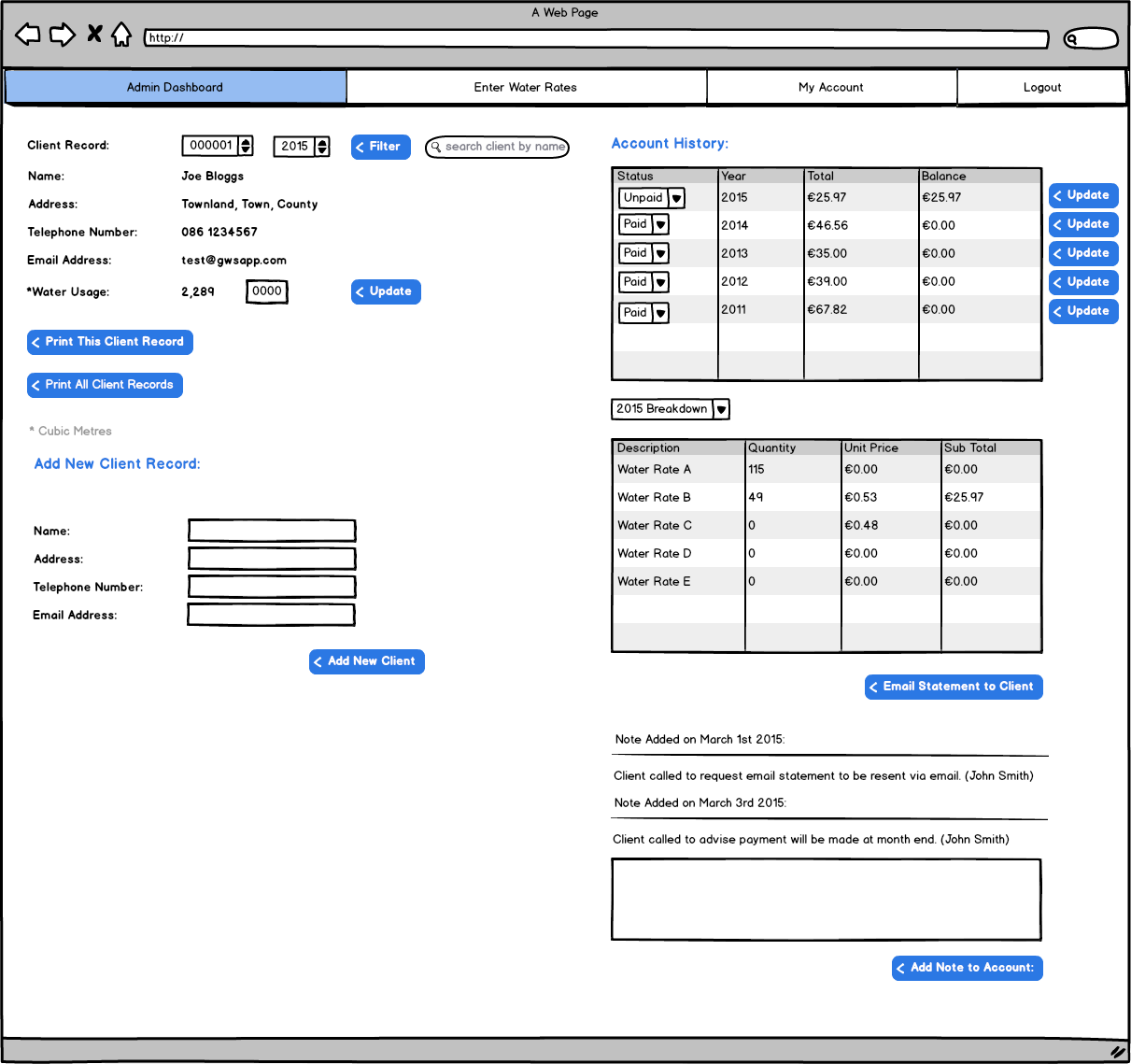
Mock-Up 2:



**Requirement 3: Graphical User Interfaces - Admin Dashboard**

**Description & Priority**

A group water scheme administrator should also be able to log in to the web-portal where they can administer the system. Only administrators will be able to see this area. It will show details of user accounts and account history. Priority on this requirement is high.

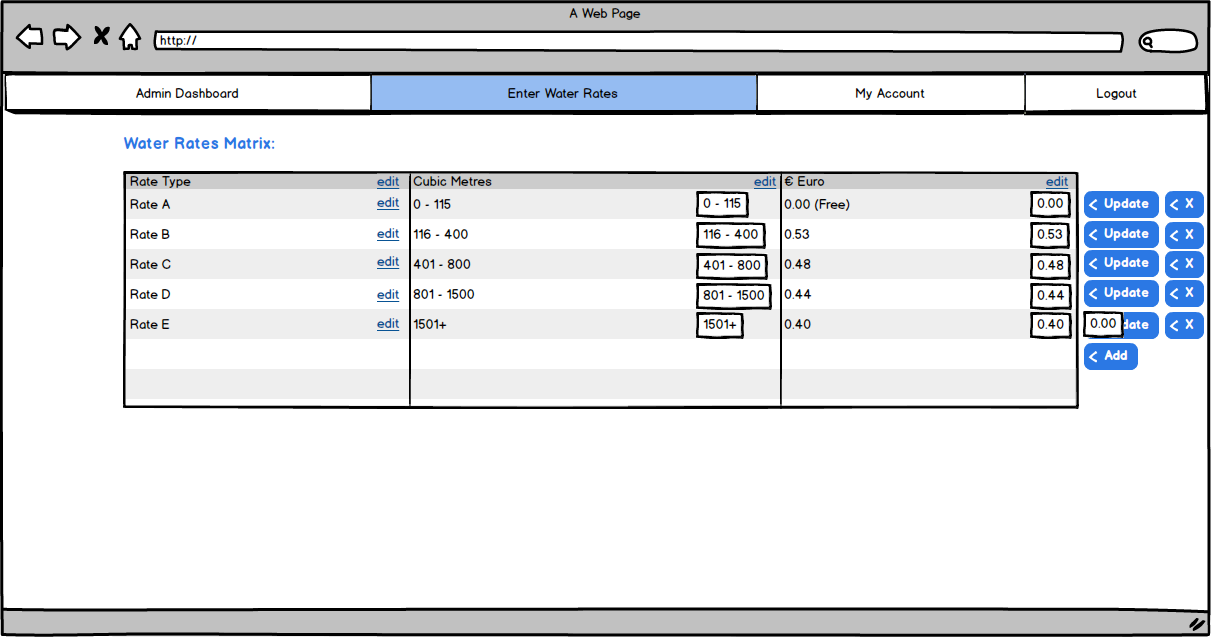
Mock-Up 3:  


**Requirement 4: Graphical User Interfaces - Water Rates (Edit)**

**Description & Priority**

A group water scheme administrator should also be able to log in to and access a tab containing the different rate bands for water fees. Visibility of rates is a medium priority requirement. The application could go live without this but it would be beneficial to have.

Mock-Up 4:



* 1. **Functional requirements**

This section includes the requirements that specify all the fundamental actions of the web application.

**Requirement 1 - Login/Logout of application**

**Description & Priority**

Users must be able to login/logout of the application using a unique username and password. A link should also be provided to be used in the case of a lost password.  Authentication is a high priority feature as the data within the application is confidential.

**Requirement Activation**

User will input their username and password into the input fields provided on the web application home page and click the login button to gain access. A pop-up box in the case of incorrect entries, prompting the user to re-enter details.

**Technical issues**

Text input fields are required on the homepage. Credentials will be verified against member records on the database back-end. Depending on whether the user is an administrator or a member they should see different menu options.

**Risks**

Unauthorised access to account information. Alteration of account information by unauthorised sources. This can be remedied by having sufficient validation measures in place to ensure that the correct authorised staff only have authorisation to make changes or access any information in the application.

**Dependencies with other requirements**

There is no dependency on other requirements as accounts will be pre-registered by the Administrator.

**Use Case Description**: A user/admin entering a username and password in order to gain access to the system.

**Primary Actor**: System Administrator/domestic user.

**Precondition**: User needs to have a pre-registered account.

**Basic flow**: User wishes to access the application to view account information.

1. User browses to the web URL.
2. User is prompted to enter his username and password and clicks login button.

**Alternate flow** :User has entered password or username incorrectly

1. User is prompted to re-enter the details again.
2. User has forgotten password.
3. User submits email address and receives link to reset password.
4. New password is created.
5. User successfully logs in.

**Requirement 2 - Filters & Searches**

**Description & Priority**

Users should be able to filter and search data at various points throughout the application. Administrators should be able to filter and search customer records. Users should be able to filter through each yearly billing period.

**Requirement Activation**

Administrators will use drop down menus and a filter button in order to request the desired output. Users will use drop down menus to view historical and current account history.

**Technical issues**

Most filtered outputs will be displayed in a tabular format. Output will be displayed as a result of a query on the relevant database table.

**Risks**

Not Applicable

**Dependencies with other requirements**

Functionality will require interaction with database tables.

**Functional Requirements**

The filter method should deliver the records contained within the parameters of the search.  This information must be viewable/obtainable in tabular form in order for it to be exported, emailed or printed.

**Use case**

Using filter/searches to obtain records.

**Scope**

The scope of this use case is to illustrate a typical process when searching through account records using either the filter or search methods.

**Description**

This use case describes the typical process when searching through account records using either the filter or search methods.

**Use Case Diagram**

To do.

**Precondition**

The system is in initialisation mode and the Admin/User is logged in.

**Activation**

This use case starts when an Admin is in the dashboard area of the UI

**Main flow**

1. The admin clicks on a filter option
2. The admin selects the client record and period they wish to view
3. The admin clicks the filter button
4. The data is returned in tabular format.

**Alternate flow**

1. The system returns no records available for that search
2. The admin re-adjusts the search settings to fit the search requirements
3. The use case continues at position 3 of the main flow

**Requirement 3 - Add a new Client Record**

**Description & Priority**

Administrators should have the ability to add additional client records to the database from the application.

**Requirement Activation**

Administrators will input client details into input fields and submit. The relevant record will be created on the database table.

**Technical issues**

**Risks**

**Dependencies with other requirements**

Describes interactions with other requirements.

**Functional Requirements**

There must be a submit button to create the record.

Records must be updated instantaneously.

There must be validation in place for missing fields or incorrectly entered email addresses.

**Use Case**: Adding a new client record

**Primary Actor:** System Administrator

Use Case Description: System Administrator has new client record to enter to system.

(If altering an existing record)

**Process flow:**

1. Admin selects update record option.

2. Admin fills the following fields with data. Name, Address, Telephone No. Email

3. Admin clicks update record and the new details are saved.

**(If creating a new record)**

1. Admin selects the add new record option

2. Admin fills the following fields with data: Name,

Address, Telephone No. Email

3. Admin clicks update record.

**Alternate flow:**

1. Admin intends to enter information and finds that the information is incomplete for new address, or name.
2. The record cannot be uploaded. These are required fields.
3. Admin obtains details.
4. Reverts to step 2 above.

**Primary Actor:** System Administrator

**Requirement 4 - Print Record(s)**

**Description & Priority**

Administrators should be able to print one or multiple client records from the admin dashboard area.

**Requirement Activation**

Administrators will select the relevant client record from a drop down and proceed to click a button to either print a) the client record being displayed or b) print all client records from the database. Or print a subset of any records, such as overdue accounts.

**Technical issues**

Must have a connected and operable printer in order for task to be successfully completed.

**Risks**

**Dependencies with other requirements**

The search and filter methods within the Admin Dashboard must be fully functional in order for successful and accurate printing of details.

**Functional Requirements**

There must be a print command. There must be a pop up box in order to select printer in the event of multiple printers.

**Use Case: Printing Customer Records**

Use Case Description: Printing customer information by a System Administrator.

Primary Actor: System Administrator/User

Precondition: System Administrator is successfully logged in and using the dashboard UI.

Basic flow:

1.System Admin/User uses the filter option to obtain a subset/all of records

2. User selects the print option, the records are then sent to the printer and printed.

Alternate flow:   
1.User uses the filter option to obtain a subset of records/selects all records  
2. Admin clicks print option and there is no printer available.   
3.User selects printer from list.  
4.Echoes #2 of Basic Flow.

**Requirement 5 - Email Statements**

**Description & Priority**

Administrators should have the ability to send an electronic statement to any clients who have provided an email address.

**Requirement Activation**

Administrators will select the relevant years account history and click the ‘email statement to client’ button in order to send a .PDF statement to them.

**Technical issues**

**Risks**

**Dependencies with other requirements**

Dependency on updating records procedure. If a valid email does not exist for a customer the action cannot be completed.

**Functional Requirements**   
Statements can be created in .pdf format for email. The records for a year can be emailed to a valid email address.A system admin should be able to complete this action from the Admin dashboard.

**Use case**

Emailing Statements to Customers

**Scope**

The scope of this use case is the emailing of  a customers account details to them.

**Description**

This use case describes the process of obtaining a customers’ account details and emailing them to the account holder.

**Use Case Diagram**

To be done.

**Flow Description**

**Precondition**

The Administrator has successfully logged in and is in the admin dashboard UI.

**Activation**

This use case starts when an  Administrator selects a selects accounts.

**Main flow**

1. The Admin selects accounts.
2. The Admin enters the customers name in the Search option.
3. Search button is clicked.
4. Admin selects year.
5. Admin selects email button to send .pdf version to customer

**Alternate flow**

1. The Admin selects accounts.
2. The Admin enters the customers name in the Search option.
3. Search button is clicked.
4. Admin selects wrong year
5. Reverts to step 2. in Main flow.

**Exceptional flow**

The Admin selects accounts.

The Admin enters the customers name in the Search option.

1. Search button is clicked.
2. Admin selects year.
3. Admin selects email button to send .pdf version to customer.
4. No email address exists for customer.
5. Message appears on screen advising of this.

**Termination**

Message appears on screen advising that the email has been sent.

**Requirement 6 - Add Note to an account**

**Description & Priority**

Administrators should have the ability to add a note to a clients account in order to provide any updates received from clients either via phone, email or otherwise.

**Requirement Activation**

Administrators will input the text into a text area and submit. It will then appear on the clients record instantly.

**Technical issues**

Possible word count issues. Also if the note is relatively large, with a few lines of text the  text may be partially ‘hidden’, with perhaps a click to expand option. This could be in place in order to keep records in a uniform format. This is something to consider during the design process and is not crucial.

**Risks**

**Dependencies with other requirements**

Describes interactions with other requirements.

**Functional Requirements**

       There must be a blank text area allocated for the administrator to update.

       There must be an update record option in order to submit the change.

**Use Case:** Updating Account

Application name: Metter Matters

Use Case Description: In the event of payment an administrator must

update the customer;s account with the demarcation paid. An administrator  can be

marking accounts paid as they trickle in throughout the business year.

Primary Actor: System Administrator

Precondition: A payment has been made and the administrator has been logged in.

Basic flow:

1. The administrator  opens the account section in the admin dashboard

2.The administrator searches for the customer through the search option.

3. The administrator adds a note to the customers account in the text area allocated.

4. The admin must click the update record button.

Alternate flow:

Termination: Message stating that the record has been successfully updated.

**Requirement 7 - Update Account Status (Paid/Unpaid)**

**Description & Priority**

Administrators should have the ability to update a clients account to display as paid or unpaid.

**Requirement Activation**

Administrators will select one of two options from a dropdown; paid/unpaid and click an adjacent ‘update’ button to update the account status as required.

**Technical issues**

            This must also be a search option which the administrator is free to obtain a list of unpaid

            members of the group scheme. This list then can also then be emailed or printed, using  a

            print option.

**Risks**

**Dependencies with other requirements**

Describes interactions with other requirements.

**Functional Requirements**

       There needs to be an option to check a box indicating ‘payment made’.

       An Update Record or Submit Changes button must exist in order for the action to be      completed.

**Use Case:** Updating Account

Application name: Metter Matters

Use Case Description: In the event of payment an administrator must

update the customer;s account with the demarcation paid. An administrator  can be

marking accounts paid as they trickle in throughout the business year.

Primary Actor: System Administrator

Precondition: A payment has been made and the administrator has been logged in.

Basic flow:

1. A list of payments has been received.

2.The administrator opens the

accounts section in the Admin Dashboard

3. The admin manually checks the box under the paid section in each a

(Those that have not been checked remain ‘unpaid’ and are available as a search

option for further querying, manipulation or transport of d ta. )

4. In the case of a change  (or  the paid box being ticked, the admin must click the update record button.

Alternate flow:

No alternate flow as both outcomes take place in the same action. Step 3

covers both eventualities (paid and unpaid)

**Requirement 8 - Edit Rates & Rate Types**

**Description & Priority**

Administrators should be able to set up rate bands. Edit existing and add new.

**Requirement Activation**

Administrators should have access via the ‘Edit Rates’ tab to edit existing rate descriptions and prices.

**Technical issues**

**Risks**

**Dependencies with other requirements**

Describes interactions with other requirements.

**Functional Requirements**

*<Create a separate use case and place it in the appendix. As an example see Appendix 7.1.>*

* 1. **Documentation requirements**

Examples of document requirements are

* How much documentation is required
* Should it be on-line, in book format or both
* To what audience is each type of documentation addressed

**Requirement 1 <name of document requirement in a few words>**

**Description & Priority**

Description of requirement and priority,

* 1. **Data requirements**

Data requirements should consider the following:

* What should the format of data be for input and output
* *How often will they be sent or received*
* How accurate must they be
* To what degree of precision must the calculations be made
* How much data flow through the system
* Must the data be retained for any period of time

**Requirement 1 <name of data requirement in a few words>**

**Description & Priority**

Description of requirement and priority,

* 1. **Non-Functional Requirements**

Specifies any other particular non-functional attributes required by the system. Examples are provided below. Remove the requirement headings that are not appropriate to your project.

**Performance/Response time requirement**

**Availability requirement**

**Recover requirement**

**Robustness requirement**

**Security requirement**

**Reliability requirement**

**Maintainability requirement**

**Portability requirement**

**Extendibility requirement**

**Reusability requirement**

**Resource utilization requirement**

System models

This section presents a more detailed description of the system model. For example DFD, ERD, UC Model etc.

System evolution

This section describes how the system could evolve over time

Appendices

* 1. **Use case 1**

**Use case**

Name of use case

**Scope**

The scope of this use case is to …….

**Description**

This use case describes the ………..

**Use Case Diagram**

Diagram should highlight actors and uses cases……..

**Flow Description**

**Precondition**

The system is in initialisation mode……..

**Activation**

This use case starts when an <Actor>…………

**Main flow**

1. The system identifies the ………….
2. The <Actor> …………...(See A1)
3. The system …………..(See E1)
4. The <Actor> ………….

**Alternate flow**

A1 : <title of A1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 3 of the main flow

**Exceptional flow**

E1 : <title of E1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 4 of the main flow

**Termination**

The system presents the next ……….

**Post condition**

The system goes into a wait state