Mobile Government
M-Government 3.0 Design Abstract
01. References

M-Gov business requirements v3.doc
M-Gov requirements specification v2.doc
Communication between providers and m-Gov gateway.doc
M-Gov_to_providers Schema.doc
Communication between Services and m-Gov gateway v3.doc
M-Gov_to_services Schema.doc
1. Background

The aim of this document is to “Design” the m-Gov gateway based on the requirements gathered from the m-Gov requirements specification document. The content of this document will specify the m-Gov infrastructure that is the design of the actual m-Gov gateway software including any required classes, data structures and modules. This document will also specify the hardware and software platforms involved, interfaces and users of the system.

As specified in the “Business Requirements Documents” and the ‘Requirements Specification Document” some of functionalities involved will be rolled out in subsequent phases of the project.

Roll out of functions in phase 1 of the project:

- all functionalities related to the m-Gov gateway that is receiving of messages, validation of messages, routing of messages, storing of messages, rendering of messages and authentication of m-Gov users.
- manual registration/deregistration of mobile numbers
- manual registration/deregistration of services and operators
- interfaces to allow posting of messages via XML
- interfaces to allow manual systems to register/deregister mobile user, create messages and view statuses of messages
- interfaces required to manage the m-Gov gateway that is to maintain services and operators (required by the support system)

Further Roll out of functions of the project:

- bulk mobile users registration/deregistration via XML from services
- reports

The following will be rolled out in subsequent phases of the project:

- direct access to appropriate data structures (such as the registration data structure) via SOAP interfaces.
- authentication – integration through the Registration & Authentication
- encryption of data via the use of certificates
- registration/Deregistration of a service via mobile phones
2. Hardware and Software platforms

As described in the business requirements document and the requirements specification document the m-Gov gateway will be deployed on the implemented web framework. This means that the back end datastructures will be implemented on the webframework's SQL servers whilst all components, ASPs, exes and schedulers on the IIS servers.

Since the m-Gov gateway is to be deployed on the Web Framework, development of all components, databases and server side scripts have to be Microsoft based that is COM+, SQL Server and ASPs respectively (to abide by the webframework policies). Since XML is to be the transport protocol (CIMU standards) between the gateway and the services and mobile operators MSXML ver 4 is to be used.

Following are the Languages and Database that the m-Gov gateway is going to be developed in:

- MSXML ver 4 and XSD
- SQL server 2000
- Visual Basic ver 6
- HTML
- ASPs
- Javascript

2.1 Schematic Overview

![Figure 1: Schematic overview of web framework. The m-Gov datastructure will reside on the SQL servers whilst the m-Gov components and ASPs will reside on the load balanced IISs.](image-url)
3. System Interfaces

Since the main function of the m-Gov project is to offer a medium between the services and the mobile operators, the interfaces involved are a vital part of the project, since it is through these interfaces that the parties involved will be able to communicate with the m-Gov gateway and hence deliver their final business function. A number of interfaces are to be offered to both the mobile operators and the services, these will consist of a number of ASP pages and a number of components exposed via SOAP. As already described the SOAP components and a few of the ASP pages (particularly those involved in registration/de-registration of mobile numbers to services and/or subjects via XML) will be delivered at a later phase of the project.

An ASP page (XML_IN.asp) will be offered to allow the mobile operator to post XML files to the gateway, whilst an ASP page (XML_MG.ASP) will be published for those services wanting to integrate with the gateway, the ASP will accept XML files from the services containing messages whilst another will be published to accept XML files required to register/de-register mobile numbers to services and/or subjects. As for the manual services a number of ASPs will be offered to allow them to register or de-register a mobile number to a service and/or subject and to send messages to appropriate registered mobile users.

Also as agreed, all services (refer to Communication between Services and m-Gov gateway v3.doc and m-Gov MoU) and operators (refer to Communication between providers and m-Gov gateway.doc and m-Gov MoU) are to publish their own Interface (in the form of a Web Page) to accept XML files from the m-Gov Gateway.

Before a service or a mobile operator can use the m-Gov services they have to be registered on the m-Gov system by the Support System, therefore a number of interfaces are to be offered to allow the support system personnel to administer the m-Gov system.
3.1 Interfaces between m-Gov Gateway and Mobile Operators

The following figure displays the messages and acknowledgements to be interchanged between the m-Gov Gateway and the operators.

![Diagram of message flow between m-Gov Gateway and Mobile Operators]

**Figure 2: Messages and acknowledgements being transferred between the operators and the m-Gov infrastructures**

If one had to view an overall picture of a message's life cycle between the m-Gov gateway and the operators one notices that there are 2 distinct generation points for a message hence 2 distinct lifecycles: one when the m-Gov infrastructure instigates (creates) a message and another when the mobile infrastructure instigates a message. One may look at the generation points as the start of a transaction.

During the lifecycle of a message a number of acknowledgements are to be forwarded to both the m-Gov and mobile infrastructures (depending on who instigates the initial message). One may view these acknowledgements as micro transactions of the original transaction.

To relate the micro transactions with their parent transaction and to distinguish between the different transactions generated by the m-Gov and mobile infrastructures a method to uniquely identify each one of the transactions (messages) is to be introduced. The only way to uniquely identify each and every transaction is to introduce a standard unique ID generation algorithm (since there exist different points of creation), whilst to identify the micro transactions a transaction type could be linked to this unique id. Using this method tractability of a message's lifecycle is guaranteed.
3.1.1 Interface Communication

As stated in the “Communication between providers and m-Gov gateway” document and “m-Gov MoU” the mobile operators are to communicate with the m-Gov gateway by posting defined XML files to an ASP page offered by the m-Gov infrastructure. Also, each mobile operator is to publish a web page to accept XML files from the gateway.

Each web page (both the m-Gov and operator’s pages) must return an object state of 200 and a string containing _ok and the file size in bytes on successfully receiving an XML file (refer to appendix 1 for code sample). If the file is not valid when checked against the XSD an object state of 200 and a string with _ERR(2) and the error message is passed.

3.1.2 Defined XSD

Given that each infrastructure will know at least the origin (either Go Mobile, Vodafone, m-Gov or the service), destination, mobile number, transaction time of each message and the actual message a hashing (SHA256) function on these values is to be generated to produce a unique identifier for the message.

Following is the defined XML structure (XSD – Refer to appendix 2 for a detailed structure). This structure encompasses all the information required for the transport of messages and acknowledgements whilst also uniquely identifying each and every message and acknowledgement.

```
<Root>
  <Transaction>
    <TransID> </TransID>
    <Orig>    </Orig>
    <Dest>    </Dest>
    <TimeStamp> </TimeStamp>
    <Type>    </Type>
    <Msg>    </Msg>
    <Mobile>  </Mobile>
    <Priority>  </Priority>
    <State>   </State>
  </Transaction>
</Root>
```

TransID Element
Unique hash value (SHA 256) computed by the service generating the message. This hash value is to be produced by hashing the concatenated string of the Origin, Destination, Mobile, Timestamp and Message.

Orig and Dest Elements
The origin and destination elements are to indicate the origin and destination of a message or acknowledgement respectively. Each one of these values is to be unique and defined beforehand. The values are 6-digit values starting of with a 5 and in the range 501500 – 501700. Following are the values assigned:
TimeStamp Element
The timestamp element indicates the time when the transaction is executed that is for example when the message is received on the mobile recipient, when the m-Gov infrastructure forwards the message, etc. The type of the timestamp element is a 19 character fixed string in the following format “dd/mm/yyyy hh:mm:ss”.

Type Element
The type element indicates if the transaction is a message, acknowledgement 1 (ack1), acknowledgement 2 (ack2) or acknowledgement 3 (ack3):

- An ack1 is instigated by the mobile operators acknowledging a message received from the m-Gov infrastructure.
- An ack2 is instigated by the mobile operators acknowledging a message received on the recipient’s mobile.
- An ack3 is instigated by the m-Gov infrastructure acknowledging a message received from the mobile operators.

Following are the values for this element:

1 - indicates that the transaction is a message
2 - indicates that the transaction is an ack1
3 - indicates that the transaction is an ack2
4 - indicates that the transaction is an ack3
101 - indicates that the transaction is a test message
102 - indicates that the transaction is a test ack

Msg Element
The msg element is to contain the actual message being passed. When the transaction is an acknowledgement this element is to be null. This element is to be of type string containing 160 characters.

Only ascii values 32 to 125 both inclusive are to be accepted except for ascii values 38 and 60 all others are to be filtered and replaced with ascii value 32 that is a white space.

Mobile Element
The mobile element is to contain the mobile number of the recipient in international format. This element is not required when acknowledgements are posted. The type of this element should be a string of 13 characters starting with 00356 (e.g 0035679123456).

Priority Element
This element is to indicate the priority or order of the message being passed to the operators. This field should be of type integer and the values passed as follows:

0 – Normal priority
1 – High priority

This element is not required when messages are sent by the mobile operators or when any type of acknowledgement is posted.
**State Element**
The status element is to indicate the status of the acknowledgement transaction that is:

- 0  –  Successful
- 1  –  Unsuccessful
- 2  –  Filtered
- others  –  Errors (refer to appendix 4 for error codes)

### 3.1.3 XML Examples

Using The Structure Proposed Above

**Scenario 1:** m-Gov sends a message to Vodafone. Assume the message originated from the law courts system:

```xml
<Root>
  <Transaction>
    <TransID>Hash1</TransID>
    <Orig>501600</Orig>
    <Dest>501501</Dest>
    <TimeStamp>06/09/2002 12:15:26</TimeStamp>
    <Type>1</Type>
    <Msg>Hello</Msg>
    <Mobile>0035699123456</Mobile>
    <Priority>0</Priority>
  </Transaction>
</Root>
```

Acknowledgement of type 1 from Vodafone to m-Gov indicating that Vodafone received the message successfully (message originated from courts).

```xml
<Root>
  <Transaction>
    <TransID>Hash1</TransID>
    <Orig>501501</Orig>
    <Dest>501600</Dest>
    <TimeStamp>06/09/2002 12:20:32</TimeStamp>
    <Type>2</Type>
    <State>0</State>
  </Transaction>
</Root>
```
Scenario 2: Go Mobile sends a message to the m-Gov infrastructure. Assume the message originated from mobile number 79123456 and the number dialed is that of the license’s service:

Message from Go Mobile to mGov. The trans_id is generated by the operator’s infrastructure.

<Root>
  <Transaction>
    <TransID>Hash2</TransID>
    <Orig>501502</Orig>
    <Dest>501602</Dest>
    <TimeStamp>06/09/2002 13:45:56</TimeStamp>
    <Type>1</Type>
    <Msg>Test</Msg>
    <Mobile>0035679123456</Mobile>
  </Transaction>
</Root>

Acknowledgement 3 from mGov infrastructure to Go Mobile’s infrastructure indicating that the message was received successfully.

<Root>
  <Transaction>
    <TransID>Hash2</TransID>
    <Orig>501500</Orig>
    <Dest>501502</Dest>
    <TimeStamp>06/09/2002 12:40:02</TimeStamp>
    <Type>4</Type>
    <Msg></Msg>
    <State>0</State>
  </Transaction>
</Root>
Scenario 3: Vodafone sends a test message to the m-Gov infrastructure.

Test Message from Vodafone to mGov. The trans_id is generated by the operator’s infrastructure, but is ignored.

```
<Root>
  <Transaction>
    <TransID>HASH</TransID>
    <Orig>501501</Orig>
    <Dest>501500</Dest>
    <TimeStamp>01/01/2008 12:00:00</TimeStamp>
    <Type>101</Type>
    <Msg>to Provider 4</Msg>
    <Mobile>0035699439212</Mobile>
    <Priority>0</Priority>
    <State>0</State>
  </Transaction>
</Root>
```

Test Acknowledgement from mGov to Vodafone.

```
<Root>
  <Transaction>
    <TransID>HASH</TransID>
    <Orig>501500</Orig>
    <Dest>501501</Dest>
    <TimeStamp>01/01/2008 12:00:00</TimeStamp>
    <Type>102</Type>
    <Msg>to Provider 4</Msg>
    <Mobile>0035699439212</Mobile>
    <Priority>0</Priority>
    <State>0</State>
  </Transaction>
</Root>
```
3.2 Interfaces between m-Gov Gateway and Services

The interfaces to be offered will allow the services to communicate with the m-Gov gateway either in a manual way that is through a number of web pages, or in an integrated way that is amalgamating use of the m-Gov functionality within the service’s applications so as to achieve the required goals.

If manual use is required by a service then sending of messages, registration/de-registration of mobile numbers to a service and/or subject and checking of message statuses’ are to be achieved via the use of a number of web pages. On the other hand acknowledgements from the m-Gov gateway and messages from citizens are to be forwarded to the services by the use of email.

If integrated use is preferred by the service than sending of messages, registration/de-registration of mobile numbers, receiving of acknowledgements and messages are to be achieved through the exchange of predefined XML files between the m-Gov gateway and the services (similar to the m-Gov to mobile operators interface described in section 3.1).

3.2.1 Registration/De-Registration of Mobile Numbers to Services and/or Subjects

One important aspect to keep in mind when registering a new mobile number is the “Data Protection Act” since the service sending the message could be sued for sending messages to recipients who are not registered for the service. Therefore one must integrate appropriate business processes to cater for checking and vetting of each and every mobile number before actual registration is computed.

Registration and de-registration is to be provided manually through the implemented m-Gov web pages and automatically using a mechanism similar to sending and receiving of messages, that is via XML (the same XSD as the one used for sending and receiving messages will be used).
3.2.2 Sending and Receiving of messages and acknowledgements

The following figure displays the lifecycle of a message and the related acknowledgements a service will receive on submitting a message.

![Diagram of message lifecycle]

**Figure 3: Messages and acknowledgements being transferred between the services, m-Gov infrastructures and operators**

When a message is instantiated from a service it is packaged with other messages into an XML file and posted to the m-Government gateway. The gateway will then validate and process the XML files forwarded (an acknowledgement per message will be posted to the service on successful processing of a message) and post the messages to the appropriate mobile operator. When the appropriate mobile operator receives the message it will then forward an acknowledgement to the m-Gov gateway stating that it successfully received the message. Another acknowledgement will also be sent to the gateway once the message is delivered to the recipient’s mobile. The acknowledgements will then be forwarded to the appropriate service.

The XML file sent by the services to the gateway must have the following structure:

```xml
<PostData>
  <Params>
    <UserID> </UserID>
    <Password> </Password>
    <Src> </Src>
  </Params>
  <Root>
    <Transaction>
      <TransID> </TransID>
      <Orig> </Orig>
      <Dest> </Dest>
      <TimeStamp> </TimeStamp>
      <Type> </Type>
      <Msg> </Msg>
      <Mobile> </Mobile>
      <Subject> </Subject>
    </Transaction>
  </Root>
</PostData>
```
Similarly the XML file sent by the m-Gov gateway to the services will have the following structure (note that the XML file might have both acknowledgements and messages):

```
<PostData>
    <Root>
        <Transaction>
            <TransID> </TransID>
            <Orig> </Orig>
            <Dest> </Dest>
            <TimeStamp> </TimeStamp>
            <Type> </Type>
            <Msg> </Msg>
            <Mobile> </Mobile>
            <State> </State>
            <Subject> </Subject>
        </Transaction>
    </Root>
</PostData>
```

**UserID Element**
User ID assigned by support system

**Password Element**
Password assigned by support system

**Src Element**
The src is required to indicate if the service using the m-Gov services is automated or manual. For all integrated services the value should always be “A”.

**TransID Element**
Unique hash value (SHA 256) computed by the service creating the message. This hash value is to be produced by hashing the concatenated string of the Origin, Destination, Mobile, Timestamp and message. This element should be made up of a string of 256 characters.

**Orig and Dest Elements**
The origin and destination elements are to indicate the origin and destination of a message or acknowledgement respectively. Each one of these values is to be unique and defined beforehand. The values are 6-digit values starting of with a 5 and in the range 501500 – 501700. Following are the values assigned:

- m-Gov – 501500
- Vodafone – 501501
- Go Mobile – 501502
- Courts – 501600
- Complains – 501601
- Licenses – 501602
- Etc – Etc

**TimeStamp Element**
The timestamp element indicates the time when the transaction is executed that is for example when the message is received on the mobile recipient, when the m-Gov infrastructure forwards the message, etc. The type of the timestamp element is a 19 character fixed string in the following format “dd/mm/yyyy hh:mm:ss”.

**Type Element**
The type element indicates if the transaction is a message or an acknowledgement:

The following acknowledgements are to be forwarded to the services:
Acknowledgement 52 (ack52) instigated once the m-Gov infrastructure acknowledges a message received from a service.
Acknowledgement 53 (ack53) instigated once the m-Gov infrastructure receives an acknowledgement that the message originally sent by the service has been received on the recipient's mobile.

Following are the values used for this element:

- 51 - indicates that the transaction is a new message
- 52 - indicates that the transaction is an ack52
- 53 - indicates that the transaction is an ack53
- 54 - indicates that the transaction is a bulk registration
- 55 - indicates that the transaction is a bulk de-registration
- 101 - indicates that the transaction is a test message
- 102 - indicates that the transaction is a test acknowledgement

**Msg Element**
The msg element is to contain the actual message being passed. When the transaction is an acknowledgement this element is to be null. This element is to be of type string containing 160 characters.

**Mobile Element**
The mobile element is to contain the mobile number of the recipient in international format. This element is not required when acknowledgements are posted. The type of this element should be a string of 13 characters starting with 00356 (e.g. 0035679123456).

**Priority Element**
This element is to indicate the priority or order of the message being passed to the operators. This field should be of type integer and the values passed as follows:

- 0 - Normal priority
- 1 - High priority

This element is not required as it is up to the gateway to assign the priority of the message depending on the service sending it.

**State Element**
The status element is to indicate the status of the acknowledgement transaction that is:

- 0 - Successful
- 1 - Unsuccessful
- others - Errors (refer to appendix 4)

**Subject Element**
The subject element is required for registration and de-registration and if needed to authenticate if a mobile number is registered to a specific subject.

It must be noted that the XML file being sent by a service **cannot** contain messages originating from another service.

The XML file posted has to be in conformance with the defined XSD (see appendix 2).

### 3.2.3 Manual Use Of The m-Gov Service

As already described, the manual services will interact with the m-Gov gateway via a number of web pages offered. A user is to logon to the system using the assigned username and password. Once authenticated, the logged on user may administer his account by changing his password, register/de-register mobile numbers to a service.
and/or subject, send messages and check the status of sent messages. A session is assigned to the user once logged on to the system so as to reduce the risk of malicious use of the system if the user does not logout when he is not using the system.

The manual systems will receive messages and all information related to any message sent via email. It is therefore up to the web master of the service to check the service’s email box for any new messages. The structure of the emails posted will be in a standard format and will include appropriately the following information (refer to figure 4 below for a sample mail):

- Transaction ID of message
- Origin Number of message
- Destination Number of message
- Type of message, i.e. an SMS or an acknowledgement
- The actual Message
- The mobile number of the recipient
- The state of the message, i.e. successful, invalid or any other error.
- The subject (if available)

![Sample Email](image)

**Figure 4:** An email received by a manual system indicating that a particular message has been successfully received on the mobile 0035699458592

### 3.2.4 Integrated Use Of The m-Gov Service

All integrated systems, using the m-Gov gateway to send and receive messages, have to communicate using XML, that is they are to post messages via a defined XML schema as previously defined (refer to appendix 3) to an ASP page (XML_MG.ASP) offered by the m-Gov infrastructure. Also, each service is to publish a web page to accept XML files from the gateway.
3.2.4.1 XML examples

**Scenario 1:** Service (e.g. Courts) sends 2 messages, one to be routed to Vodafone and the other to Gomobile:

Messages from Courts to Vodafone and Gomobile. Assume that ID of courts is 500601 and ID of Vo and Go are 500501 and 500502 respectively. The trans_id is generated by service:

```
<PostData>
  <Params>
    <UserID>burdv001</UserID>
    <Password>burdv100</Password>
    <Src>A</Src>
  </Params>
  <Root>
    <Transaction>
      <TransID>000000000002</TransID>
      <Orig>501601</Orig>
      <Dest>501501</Dest>
      <Type>51</Type>
      <Msg>Msg no 1 Hello</Msg>
      <Mobile>0035699000002</Mobile>
      <Subject>Case0001</Subject>
    </Transaction>
    <Transaction>
      <TransID>000000000003</TransID>
      <Orig>501601</Orig>
      <Dest>501502</Dest>
      <TimeStamp>20/09/2002 11:37:35</TimeStamp>
      <Type>51</Type>
      <Msg>Msg no 2 Hello</Msg>
      <Mobile>0035679000001</Mobile>
    </Transaction>
  </Root>
</PostData>
```

In the above example the courts application is posting two messages to the gateway. The first message contains a Subject whilst the second does not. This means that when the first message is validated the mobile number provider must be registered to the given subject and service. In the second message the subject is not present therefore the mobile number is just validated against the service.

**Scenario 2:** m-Gov gateway forwards 1 message to a service (e.g. Courts):

Messages from mGov to Courts. Assume that ID of courts is 500601 and ID of Vodafone and Go Mobile are 500501 and 500502 respectively. The trans_id is generated by the mobile provider:

```
<PostData>
  <Root>
    <Transaction>
      <TransID>bje360hbdhehkt6yu</TransID>
      <Orig>501501</Orig>
      <Dest>501601</Dest>
      <TimeStamp>20/09/2002 11:37:45</TimeStamp>
      <Type>51</Type>
      <Msg>Test Message</Msg>
      <Mobile>0035679000002</Mobile>
    </Transaction>
  </Root>
</PostData>
```
In the above XML file the m-Gov gateway posts a message to the courts application.

**Scenario 3:** m-Gov gateway sends an acknowledgement to a service (e.g. Courts) indicating that a message was successfully processed by the gateway.

```xml
Messages from mGov to Courts. Assume that ID of courts is 501601. The trans_id was originally generated by the service:

<PostData>
  <Root>
    <Transaction>
      <TransID>078ygf0000000000</TransID>
      <Orig>501500</Orig>
      <Dest>501601</Dest>
      <Type>52</Type>
      <State>0</State>
    </Transaction>
  </Root>
</PostData>
```

In the above XML file the m-Gov gateway posts an acknowledgement message to the courts application indicating that the particular message was successfully processed by the gateway. The type of the message is 52.

**Scenario 4:** m-Gov gateway sends an acknowledgement to a service (e.g. Courts) indicating that the mobile recipient successfully received the message.

```xml
Messages from mGov to Courts. Assume that ID of courts is 500601 and ID of Vodafone and Go Mobile are 500501 and 500502 respectively. The trans_id was originally generated by the service:

<PostData>
  <Root>
    <Transaction>
      <TransID>000000000asdf</TransID>
      <Orig>501501</Orig>
      <Dest>501601</Dest>
      <Type>53</Type>
      <State>0</State>
    </Transaction>
  </Root>
</PostData>
```

In the above XML file the m-Gov gateway posts an acknowledgement message to the courts application indicating that the particular message was successfully received (state = 0) on the mobile of the recipient. The type of the message is 53.
Scenario 5: m-Gov gateway sends an acknowledgement to a service (e.g. Courts) indicating that the mobile recipient did not received the message.

Messages from mGov to Courts. Assume that ID of courts is 500601 and ID of Vodafone and Go Mobile are 500501 and 500502 respectively. The trans_id is generated by the mobile provider:

```xml
<PostData>
   <Root>
      <Transaction>
         <TransID>58gdhtgfdtgd00asdf</TransID>
         <Orig>501501</Orig>
         <Dest>501601</Dest>
         <TimeStamp>20/09/2002 11:51:35</TimeStamp>
         <Type>53</Type>
         <State>1</State>
      </Transaction>
   </Root>
</PostData>
```

In the above XML file the m-Gov gateway posts an acknowledgement message to the courts application indicating that the particular message was unsuccessfully received (state = 1) on the mobile of the recipient. The type of the message is 53.

Scenario 6: m-Gov gateway sends an acknowledgement to a service (e.g. Courts) indicating that the message generated contains an error (the mobile recipient is not registered in the centralized database).

Messages from mGov to Courts. Assume that ID of courts is 500601 and ID of Vodafone and Go Mobile are 500501 and 500502 respectively. The trans_id is generated by the mobile provider:

```xml
<PostData>
   <Root>
      <Transaction>
         <TransID>078ytgf0000000000</TransID>
         <Orig>501500</Orig>
         <Dest>501601</Dest>
         <Type>52</Type>
         <State>116</State>
      </Transaction>
   </Root>
</PostData>
```

In the above XML file the m-Gov gateway posts an acknowledgement message to the courts application indicating that the particular message was unsuccessfully processed by the gateway since the mobile number is not registered to receive messages for that particular service (state = 116). The type of the message is 52.
3.3 Support System Interfaces

As discussed above the interface offered will allow the support system to register, deregister and maintain services and mobile operators whilst also register, deregister and maintain users of the services.

3.3.1 Registration, deregistration and maintenance of services

Registrations
Any service requiring use of m-Gov services is to apply and register with the implemented Support System. Once the service is registered it is provided with the following information:

- Unique identification number dialled by mobile users to communicate with the service (issued by the Malta Communication Authority)
- Service’s email address

It is imperative that apart from giving the usual registration information, such as the user responsibilities for the service, password, etc, the service’s registration must indicate the interface mechanism with the m-Gov gateway (i.e. Manual or Integrated).

Deregistrations
If a service no longer requires use of the m-Gov services it could be deregistered. With this functionality the appropriate service is set to de-activated and not deleted, whilst users linked to the service are unlinked.

Maintenance
If maintenance on a service is required such as changing the email address this is to be done through these interfaces.

3.3.2 Registration, deregistration and maintenance of users

Registrations
Any user required to use the m-Gov services (that is to send messages) has to apply and register with the implemented Support System. Once the user is registered he/she is provided with the following information:

- Username
- Password

Two distinct roles exist “Administrator” and “Message Sender”. This has to be specified on registration since an Administrator apart from sending messages could register and deregister mobile numbers to the service and/or a subject whilst a Message Sender user can only send messages.

De-registrations
Used to de-register a service user. With this functionality the appropriate user is set to de-activated and the appropriate links to other services are also deleted.

Maintenance
If maintenance on a user is required such as changing the user name this is to be done through these interfaces.

4.5 Web Page Navigation Paths
Following are sample diagrams of the web pages to be offered by the m-Gov system so as to allow manual services and the support system to interact with it.

On accessing the above m-Gov URL a login screen (figure 10) will be displayed. The user is to enter his user id and password on this screen in order to access the m-Gov system.

![m-Gov Login](image)

**Figure 10: m-Gov Logon screen**

On inserting a valid user ID and password the user is prompted with an appropriate “Main Screen” depending on the role of the user (refer to section 5 for the different roles available for the m-Gov system):

- If a user has a systems administrator role than he/she is prompted with a screen as seen in figure 11. The user will have rights to register new m-Gov services, de-register m-Gov services, edit m-Gov services, create authorised users, modify authorised users and modify roles of exiting authorised users.

- If the user has administrator and message sending rights than he/she is prompted with a screen as seen in figure 12. The user will have rights to register and de-register mobile numbers to a service, create or delete subjects for a service, register and de-register mobile numbers to subjects and send SMS messages.

- If the user has administrator rights only he/she will be prompted with a screen as seen in figure 13. The user will have rights to register and de-register mobile numbers to a service, create or delete subjects for a service and register and de-register mobile numbers to subjects.

- If the user has message sending rights only he/she is prompted with a screen as seen in figure 14. The user will only have rights to send SMS messages.
**Figure 11:** *Entry Screen for a system administrator*

**Figure 12:** *Entry Screen for an administrator and message sender*

**Figure 13:** *Entry Screen for an administrator only user (send message option is disabled)*
**Figure 14**: Entry Screen for a message sender user (the user can only send SMS messages)
5. User Groups

Following are the user roles used in the m-Gov system. One must take note that a user might have more than one role assigned to him per service and that a user might have different roles on many services for example, a person might be a message sender for one system whilst a service administrator and a message sender on another system:

5.1 System Administrator

System administrator role will be given to all the Call Centre employees. This role empowers the user to create, maintain and deactivate services and users as well as link users to services. A system administrator role on the other hand does not have the right to send messages and create and link mobile user to services.

5.2 Message Sender

A message sender role will be assigned to all users (including applications) requiring functionality to send messages. This role will not allow users to assign mobile numbers to services.

5.3 Service Administrator

The service administrator role will be assigned to users assigned to create, maintain and deregister mobile users to a service. This role does not allow the user to send messages.

5.4 Message Sender and Service Administrator

A user with this role would be able to send messages as well as administer a service. This role is a combination of a message sender role and a service administrator role.
6. Appendices

6.4 Appendix 4 – Error Codes

Following are the errors that could be forwarded by the gateway:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR_OK</td>
<td>0</td>
</tr>
<tr>
<td>ERR_Failed</td>
<td>1</td>
</tr>
<tr>
<td>ERR_Filtered</td>
<td>2</td>
</tr>
<tr>
<td>ERR_Unhandled</td>
<td>11</td>
</tr>
<tr>
<td>ERR_UnhandledDB</td>
<td>12</td>
</tr>
<tr>
<td>ERR_CloseSesFailed</td>
<td>13</td>
</tr>
<tr>
<td>ERR_InsertJournal</td>
<td>14</td>
</tr>
<tr>
<td>ERR_Internal</td>
<td>15</td>
</tr>
<tr>
<td>ERR_LogInvalid</td>
<td>100</td>
</tr>
<tr>
<td>ERR_SrvInvalid</td>
<td>101</td>
</tr>
<tr>
<td>ERR_ConnectionFailed</td>
<td>102</td>
</tr>
<tr>
<td>ERR_RecordsetFailed</td>
<td>103</td>
</tr>
<tr>
<td>ERR_NoData</td>
<td>104</td>
</tr>
<tr>
<td>ERR_ClientLoggedIn</td>
<td>105</td>
</tr>
<tr>
<td>ERR_AccountDisabled</td>
<td>106</td>
</tr>
<tr>
<td>ERR_XMMLoadFailed</td>
<td>107</td>
</tr>
<tr>
<td>ERR_AccessDenied</td>
<td>108</td>
</tr>
<tr>
<td>ERR_SrvNotActive</td>
<td>109</td>
</tr>
<tr>
<td>ERR_SrvNoEmail</td>
<td>110</td>
</tr>
<tr>
<td>ERR_DataTypeNotValid</td>
<td>111</td>
</tr>
<tr>
<td>ERR_OpenConnFailed</td>
<td>112</td>
</tr>
<tr>
<td>ERR_HashExpired</td>
<td>113</td>
</tr>
<tr>
<td>ERR_NoXMLData</td>
<td>114</td>
</tr>
<tr>
<td>ERR_XMLValidFail</td>
<td>115</td>
</tr>
<tr>
<td>ERR_NodeInvalid</td>
<td>116</td>
</tr>
<tr>
<td>ERR_SrvNoAddress</td>
<td>117</td>
</tr>
<tr>
<td>ERR_ImportFailed</td>
<td>118</td>
</tr>
<tr>
<td>Public Const ERR_AllRecsFailed</td>
<td>119</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>Public Const ERR_MGovSessionFailed</td>
<td>120</td>
</tr>
<tr>
<td>Public Const ERR_NewOutBoxMsgFailed</td>
<td>121</td>
</tr>
<tr>
<td>Public Const ERR_GetJournalFailed</td>
<td>122</td>
</tr>
<tr>
<td>Public Const ERR_NoRecInJournal</td>
<td>123</td>
</tr>
<tr>
<td>Public Const ERR__UserAlreadyRegistered</td>
<td>151</td>
</tr>
<tr>
<td>Public Const ERR__UserNotRegistered</td>
<td>152</td>
</tr>
<tr>
<td>Public Const ERR__SubjNotActive</td>
<td>153</td>
</tr>
<tr>
<td>Public Const ERR__UserActiveFail</td>
<td>154</td>
</tr>
<tr>
<td>Public Const ERR__MobileParamNA</td>
<td>155</td>
</tr>
<tr>
<td>Public Const ERR__SubjectInvalidParent</td>
<td>156</td>
</tr>
<tr>
<td>Public Const ERR__UserNotRegisteredSrv</td>
<td>157</td>
</tr>
<tr>
<td>Public Const ERR__UserNotRegisteredSubj</td>
<td>158</td>
</tr>
</tbody>
</table>