



USSD

User Guide

This document is aimed at

Grapevine Affiliate who have been provisioned to use USSD

Version

1.0

Document location

On Owl at /Documents/Projects/USSD/USSD_User_Guide_V1_0.docx

Date

07 June 2010

Support

For assistance, please contact Grapevine: +27 21 702-3333 or email info@vine.co.za.

Feedback

Was this document useful? What can we do to make this user guide more useful and informative?
Please email info@vine.co.za with your comments and feedback.

Table of Contents

1. Introduction.....	4
1.1. What is the purpose of this document?	4
1.2. What is USSD?.....	4
1.3. USSD example	5
1.4. Why use USSD?	5
1.5. Common applications of USSD technology	5
2. How does Grapevine use USSD technology?	6
2.1. The process of a message over a network.....	6
2.2. How does Grapevine’s USSD interact with a client application?.....	7
2.2.1. USSD prompts method	7
2.2.2. The USSD gateway method	8
3. USSD concepts	9
3.1. About a USSD session	9
3.2. About prompts and responses.....	9
3.3. About USSD strings	9
3.4. About message types.....	10
3.5. About parameters.....	10
3.6. About response types	10
3.7. About the XML definition file.....	11
3.8. About request and response XMLs.....	11
4. Communicating with USSD – request and response XMLs.....	12
4.1.1. Initial request XML <ussdAppRequest>	12
4.1.2. Initial response XML <ussdAppResponse>	13
4.1.3. Action request XML <ussdAppRequest>.....	13
4.1.4. Action response XML <ussdAppResponse>	13
4.1.5. Send aborts request XML <ussdAppRequest>	14
5. Appendix A – additional examples.....	14
5.1. Bank account balance request.....	14
5.2. Top up pre-paid account balance request.....	15
5.3. Regional weather request.....	17
5.4. Competition entry.....	18

Abbreviations

GSM	Global System for Mobile Communication
GVI	Grapevine Interactive
IM	Instant Messaging
NI USSD	Network Initiated Unstructured Supplementary Service Data
SMS	Short Messaging Service
USSD	Unstructured Supplementary Service Data
WAP	Wireless Application Protocol
WASP	Wireless Application Service Provider
XML	eXtensible Markup Language
XSD	eXtensible Schema Definition

1. Introduction

1.1. What is the purpose of this document?

The purpose of this document is to:

- describe Grapevine's USSD functionality
- document the XML files used in the USSD application so that an Affiliate understands how they will need to code their application (if not coded by Grapevine) to receive data from and send data to USSD
- provide examples of the way Grapevine's USSD can be used by clients so that an efficient campaign can be developed

1.2. What is USSD?

Unstructured Supplementary Service Data (USSD) is a technology used to facilitate the transmission of multiple text messages between a mobile phone and a client application program across a mobile phone network. Its main feature is that it enables **interactivity** between a mobile phone user and a client application, at high speed and in real time.

USSD appears similar to Short Messaging Service (SMS), but USSD is session-based, rather than transaction-based, and multiple USSD messages are transmitted during a session. To understand the difference between USSD and SMS, USSD can be compared to Instant Messaging (IM) which is immediate and continuous until broken by one of the communicating parties, while SMS can be compared to email, which is a single, once-off communication.

USSD messages, unlike SMSs, are not stored and forwarded, or sent to a phone's Inbox. USSD messages are displayed directly, and immediately, on a phone's screen when they are sent during a USSD session. See [3.1 About a USSD session](#).

The messages sent in a USSD session comprise prompts from an application and responses by a user. See [3.2 About prompts and responses](#).

Note: USSD messages can contain up to 160 characters, including spaces, just like an SMS.

1.3. USSD example

One of the most common uses for USSD is to make it easy for a prepaid mobile phone user to query a prepaid account balance or top up a prepaid account. If a user wants to top up their mobile phone account, they can send a prepaid voucher number ('1234567890' in the example below) to a USSD string ('141' in the example below) and will receive a reply from their network that their account has been topped up. Together, these transactions are called a USSD session. It is opened when the request is made and closed when the response is sent.

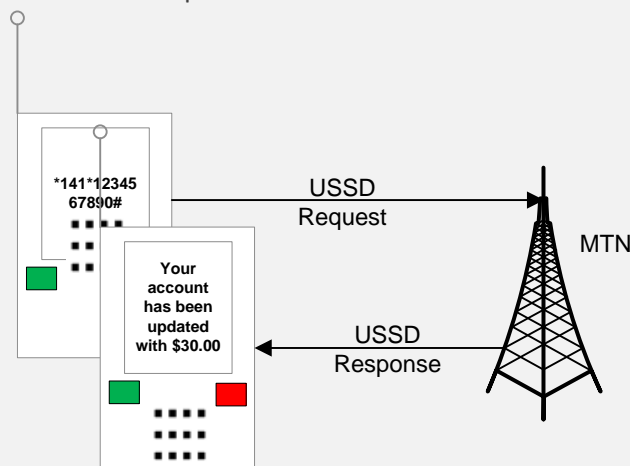


Figure 1 – USSD session

The above example is a simple one; USSD really comes into its own when multiple transactions are sent between client and user, resembling a conversation. See [Appendix A – additional examples](#).

1.4. Why use USSD?

- USSD allows client – customer interactivity with very little delay between request and response.
- USSD transactions are more cost-effective than SMS transactions because USSD transactions are typically billed at a low per second rate as long as a session is open, rather than per transaction as is the case with SMS.
- USSD technology is supported by most GSM mobile phones and no extra software is needed to make use of it.
- It can be used effectively as a marketing aid and is less invasive than telemarketing.
- USSD lends itself to being used for applications like weather services, hotel reservation services, etc. because a user's requirements can easily be ascertained by prompting them to make menu selections. See [5.3 Regional weather request](#).

1.5. Common applications of USSD technology

USSD is most commonly used to make it easy for a prepaid mobile phone user to query a prepaid account balance or top up a prepaid account. USSD is also well suited to an application that requires a user makes a selection from a menu.

But USSD has many other uses:

- Mobile chatting
- 'Please call me' requests
- Campaigns and competitions

- Surveys
- Subscription services
- Downloads
- Enquiries
- Remote transactions
- Browsing information via text menus
- Services, e.g. online gambling
- Reservation services

2. How does Grapevine use USSD technology?

2.1. The process of a message over a network

Grapevine’s USSD transmits multiple text messages between a mobile phone and a client application program using XML files. The transmission of these messages comprises a series of prompts (requests) and responses (see [Figure 7 – Requests and responses](#)).

6 XML files facilitate communication between USSD and the client application.

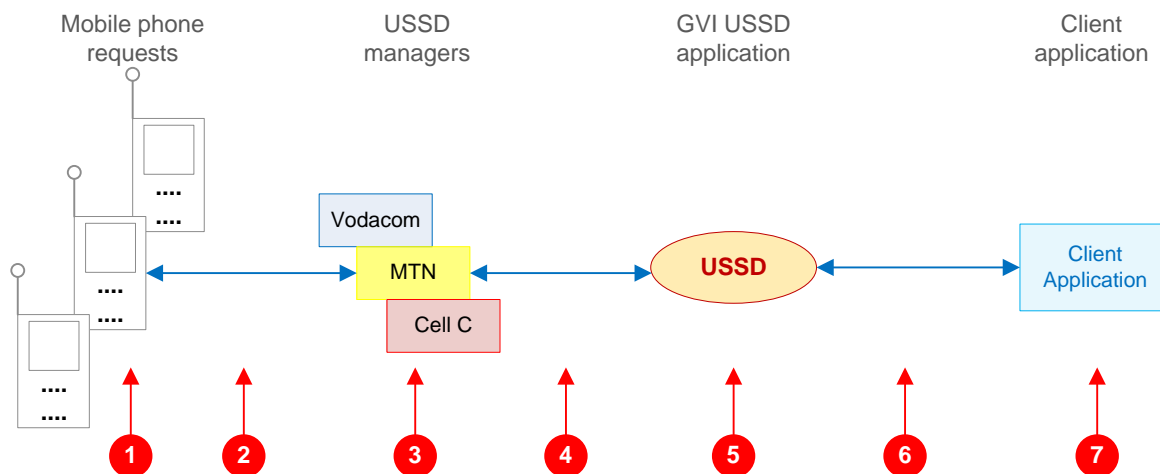


Figure 2 – The passage of a message over a network

1. A USSD session is activated by a mobile phone user typing a string of characters comprising an asterisk (*), followed by a number of digits and ending with a hash (#), e.g. *120*127#. A mobile phone handset recognises these characters as a **USSD string**, and uses a USSD bearer or manager to transmit a text message request to a client application instead of making a voice call.
2. The initial request by the mobile phone user is first routed to a mobile phone operator, like MTN, Vodacom or Cell C. The request is then routed via the network operator’s USSD manager to Grapevine’s USSD application and then to a client application program, which, in turn, sends a response back to the mobile phone.
3. Because a large number of users may be accessing USSD at the same time, the network operators’ USSD managers will create unique session IDs for each session, so that as messages stream back and forth, the session they belong to can be identified. The session ID is one of the parameters the client application will always receive from USSD in an XML file. (Another is the MSISDN.)
4. Every session begins with what is called the initState. This initial state determines where to route the **first** message.
5. Each request and response in a USSD session is called a transaction. USSD inspects these transactions and routes them to the client application, or checks the validation XML file for a new prompt, e.g. a sub-menu, to be sent to the user.

6. When it is required to send user data to the external client application, Grapevine performs an HTTP post of name / value pairs. If the client application has not been developed by Grapevine, it may be necessary for the data sent by the USSD application to be translated into a format recognisable by the client application.

An example of a name/ value pair would be <name>idnumber</name>/ <value>7210120176081</value>. The name/ value pairs would correspond to the data being transmitted between client application and user using XML files.

7. The message sent to a client application will require a response which will be routed to the user via USSD.

Note: A client application may be a custom application written by Grapevine for a client or may be an application a client already has. Grapevine writes the XML that will be used by all client applications. You, the client, need to know the format of the XML which you will receive from USSD and the format of the XML which you need to send back to USSD. See [4 Communicating with USSD – request and response XMLs](#).

2.2. How does Grapevine’s USSD interact with a client application?

There are two ways that Grapevine’s USSD can interact with a client application. Which method is used depends on the extent to which you, the client, wish to retain control over the messages sent to the mobile phone user.

We will use the same example, requesting an account balance, to demonstrate the two methods.

2.2.1. USSD prompts method

In the **USSD prompts method**, USSD interacts with a mobile phone user by sending prompts that it reads mainly from a pre-populated XML file. Examples of prompts include a Welcome message asking the user what he would like to do, or a menu from which a user must make a selection.

USSD then analyses the mobile phone user’s responses and responds by either sending another prompt, perhaps requesting more information, to the user or by sending the user’s response to a client application which analyses it. This method allows USSD to do a lot of the hard work and makes the transmission of messages very fast because it is only when the client database needs to be accessed that the client application is called.

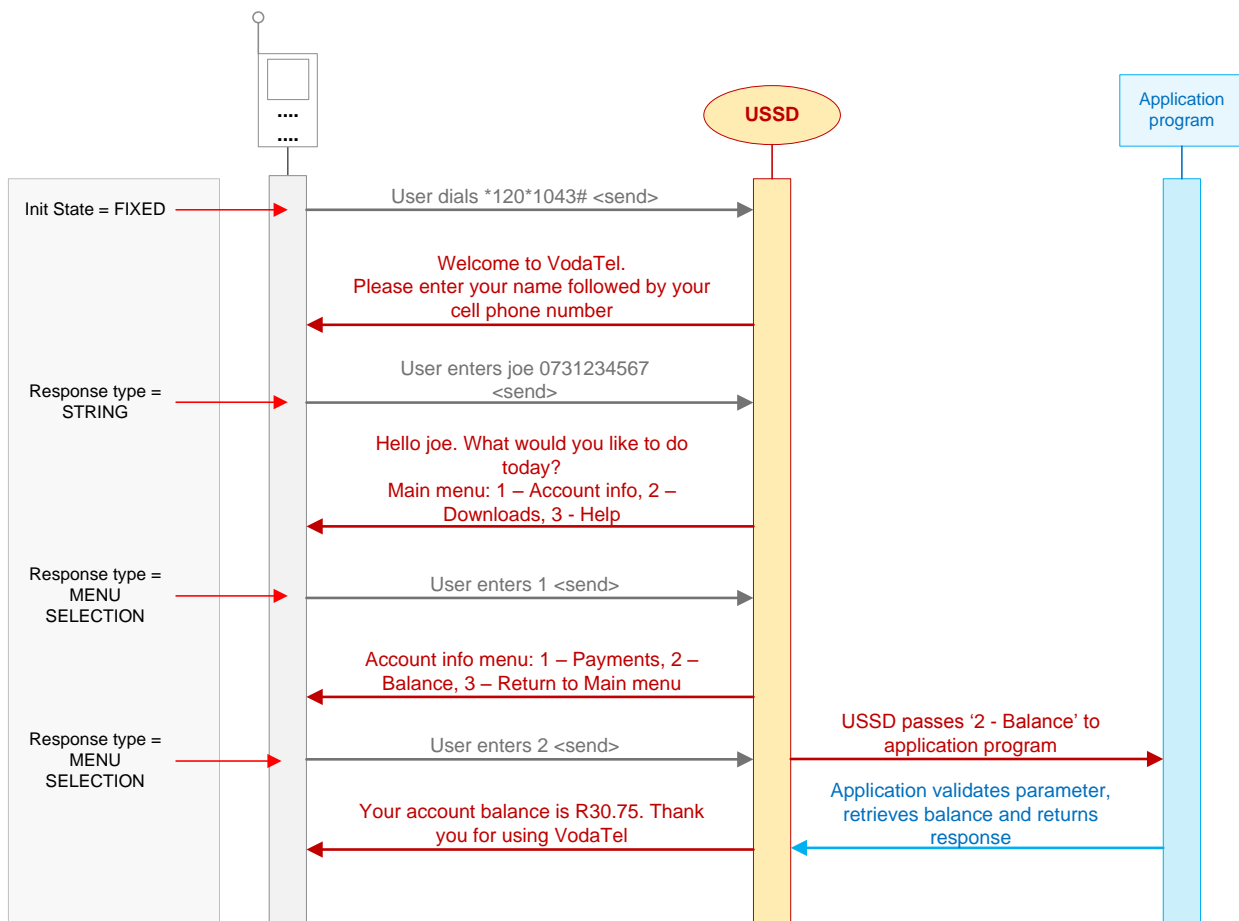


Figure 3 – requesting an account balance – USSD prompts method

1. A USSD session is initiated by a user entering a string of symbols and digits on a mobile phone, e.g. ***120*1043#**. In our example, this string is a request for a pre-paid mobile phone account balance.
2. The USSD application responds by sending a ‘Welcome ...’ message back to the user in a text string, asking for his name and mobile phone number. This is called a prompt.
3. The user enters his name and mobile phone number. This is called a response.
4. USSD now sends a text menu prompt to the user.
5. The user selects a menu option, e.g. 1 – Account info.
6. USSD now sends a sub-menu prompt to the user.
7. The user selects a menu option, e.g. 2 – Account balance.
8. It is only at this point that the USSD application passes the menu selection response to the client application, using an XML file.
9. The client application extracts the user’s account balance from its database and sends this balance back to USSD using an XML file.
10. The USSD application sends this balance to the user’s mobile phone.

2.2.2. The USSD gateway method

USSD can also facilitate the interaction between a client application and a mobile phone user by operating purely as a gateway. In the USSD gateway method, USSD simply passes transactions between the client and the phone user without analysing them, i.e., it does not itself interpret responses by the user and determine at what stage the client application should be invoked. This scenario is most often used when a client writes their own application and wishes to control all transactions completely. Let’s see how this would work if we used the same example as we did above.

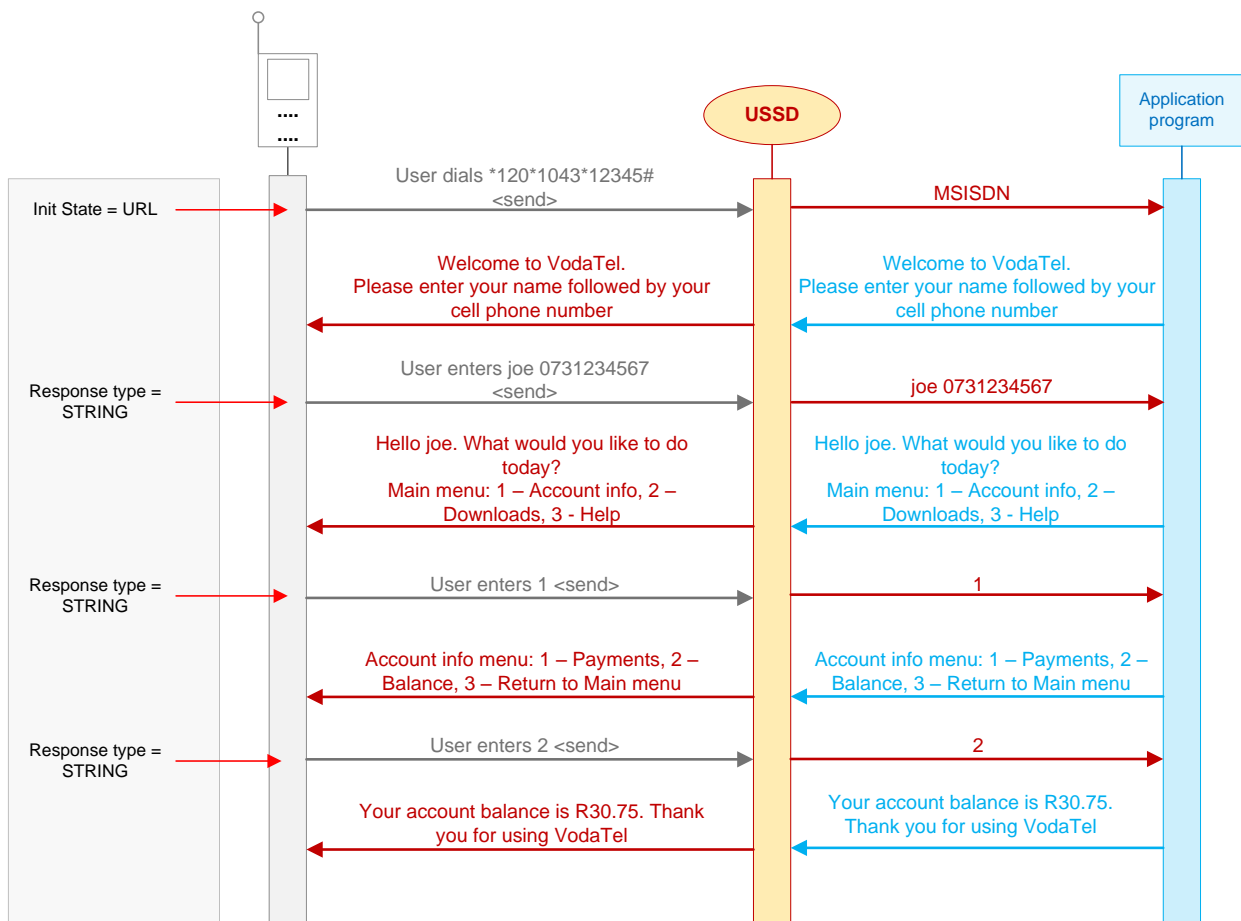


Figure 4 – requesting an account balance – USSD gateway method

3. USSD concepts

3.1. About a USSD session

A USSD session starts when a mobile phone user sends a USSD string request to an application and ends when the communication between the two is completed or there is a timeout. See [3.3 About USSD strings](#). When there are numerous requests and responses between the two, the communication resembles a real-time conversation. See [5 Appendix A – additional examples](#).

3.2. About prompts and responses

Message transmission terminology can be confusing. When we refer to a message sent from a mobile phone to USSD, we use the term ‘response’ except for the original process-initiating message. When we refer to a message sent from the USSD application to the mobile phone user, we use the term ‘prompt’ (except for the final process-terminating message). The reason for this is that we are continuously trying to ‘prompt’ a ‘response’ from the user.

3.3. About USSD strings

USSD originating requests always include a USSD string. A USSD string comprises an asterisk (*) followed by a number of digits and ends with a hash (#), e.g. *120*127#. A USSD session is initiated by the transmission of a unique USSD string that identifies the request and enables the network operators to

route the request so it can be processed by the correct WASP. USSD strings are provided to a WASP, like Grapevine, by the network operators and each string is reserved for an specific application or request.

Note: Many people will recognise one of the USSD strings to request an account balance from the network operator MTN: *141#. Grapevine's USSD strings often use two asterisks and a hash. For instance, a request to enter a particular competition may look like this: *2020*1535#. A USSD string with a parameter, e.g. a voucher number, may look like this: *2020*1535*29283928#

3.4. About message types

There are three types of session messages:

- **Originating message** – an originating message is the first message sent (always initiated in the Grapevine process by a mobile phone user) to start a session and always contains a USSD string, e.g., *127*2020*1575678#. '1575678' is a parameter; this indicates that USSD needs to send the request to the client application.
- **Data messages** – a series of requests and responses that comprise the text prompts and responses between a mobile phone and a client application, or a mobile phone and the USSD application.
- **Terminating message** – a terminating message ends a session and transports the final piece of content that may be sent by the client application, e.g., **Thank you for entering this competition.** If a mobile phone user simply switches off their phone, an 'Abort' terminating message will be sent by the USSD manager that is managing the session. If the session times out, the message will be 'Timeout'. Alternatively, the terminating message will be triggered by a keyword in the XML that is recognised by the USSD application, e.g., 'End'. In the latter case, the session has come to a 'natural' end.

3.5. About parameters

An originating message is the first message sent (always initiated in the Grapevine process by a mobile phone user) to start a session and always contains a USSD string, e.g., *127*2020*1575678#. '1575678' is a parameter (for example, a prepaid voucher number); this indicates that USSD needs to send the request to the client application.

3.6. About response types

A USSD session comprises a series of requests and responses moving between mobile phone, network operator, USSD and client application. You need to understand the two types of message response that a user can make because when you design your service, how the definition XML file is configured will depend on what type of response you are expecting from the user.

There are two response types:

- **Menu type** response – A menu is a set of options presented to a user. A menu selection is the option a user chooses. So, a menu type response is always a menu selection.

Menu	Menu selection response example
Hello joe. What would you like to do today? Main menu: 1 – Account info 2 – Downloads	1

3 – Help	
----------	--

Figure 5 – menu type response

- **String type** response – A string is comprised of a set of characters that can also contain spaces and numbers. To all intents and purposes, it can contain anything. A string is usually elicited from a user as a response to a question or request for information. So, a string type response is always a string of characters and spaces.

Request	String response example
Welcome to VodaTel. Please enter your name followed by your mobile phone number	joe 0731234567

Figure 6 – string type response

Note: USSD messages can contain up to 160 characters, including spaces, just like an SMS.

3.7. About the XML definition file

There are six XML files in USSD. The first USSD XML file, referred to as the **definition XML**, is used by the USSD application program to determine what text to send to the mobile phone after the user has initiated a USSD session. It also determines the route of the next message, for instance, to the mobile phone user, to the client application or to close the session.

The other five XML files transport the requests and responses passed between USSD and the client application. See [3.8 About request and response XMLs](#). Grapevine writes the XML definition file.

3.8. About request and response XMLs

There are six XML files in USSD. The first USSD XML file, referred to as the **definition XML**, is used by the USSD application program to determine what text to send to the mobile phone after the user has initiated a USSD session. It also determines the route of the next message, for instance, to the mobile phone user, to the client application or to close the session.

The other five XML files transport the **requests** and **responses** passed between USSD and the client application. To view the format of the request and response XML files, see [4 Communicating with USSD – request and response XMLs](#).

- USSDAppRequest (initial request)
- USSDAppResponse (initial response)
- USSDAppRequest (subsequent request/s)
- USSDAppResponse (subsequent response/s)
- USSDAppRequest (abort request – no response is returned for this request)

The XML files are always created for clients by Grapevine. Grapevine can create the back-end application as well, or a client can create their own back-end applications. What clients need to know is the format of the XML files they will receive from USSD (containing the data input by the user) and the format of the XML files they will need to send to USSD (containing the prompts to the user).

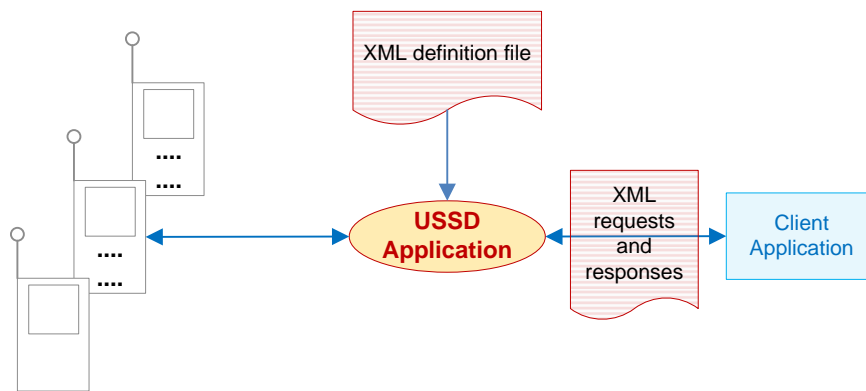


Figure 7 – Requests and responses

4. Communicating with USSD – request and response XMLs

There are three possible request XMLs that may be sent to an Affiliate from USSD and two response XMLs that an Affiliate will need to send to USSD. An Affiliate will need to understand the format of these files.

- **Initial request XML** (the first request sent by USSD to the client application)
- **Action request XML** (these are the requests that are sent by USSD to the client application except for the first request)
- **Initial response XML** (the first response sent by the client application to USSD)
- **Action response XML** (these are the responses that are sent by client application to USSD except for the first response)
- **Abort request XML** (a request sent to the client application when the service provider notifies USSD the session has timed out or been aborted by the user)

Now, let's take a look at the XML files that you can expect to receive from Grapevine and the format of the response XML files you need to reply with.

Note: An XML file only describes and transports text; it does not define the format of the text or validate it. This is done by the application itself.

4.1.1. Initial request XML <ussdAppRequest>

Below is a sample of the first request XML file that will be sent to the client application:

```

<ussdAppRequest>
  <type>initial</type>
  <msisdn>27731234567</msisdn>
  <sessionId>721082474</sessionId>
  <network>MTN</network>
  <parameter>Yes</parameter>
</ussdAppRequest>
  
```

Notes:

- <type> is always 'initial' for the first request.
- <parameter> will include user input, if any. It may be a menu selection or a string.
- <network> can be MTN, Vodacom or Cell C.

4.1.2. Initial response XML <ussdAppResponse>

Below is a sample of the first response XML file that the client application will need to send back to USSD in response to USSD's first request:

```
<ussdAppResponse>
  <prompt>Welcome to Vodatel. Please enter your name followed by your cell phone
  number</prompt>
  <state>name of the initState</state>
</ussdAppResponse>
```

Note: <state> indicates to USSD where in the definition XML it needs to go next. For instance, the client application could validate user input and point USSD to an error routine.

4.1.3. Action request XML <ussdAppRequest>

Below is a sample of a request XML file (after the first one) that will be sent to the application:

```
<USSDAppRequest>
  <type>action</type >
  <msisdn>27731234567</msisdn>
  <sessionId>721082474</sessionId>
  <name>accountbalance</name>
  <fields>
    <field>
      <name>account</name>
      <value>2</value>
    </field>
  </fields>
</USSDAppRequest>
```

Notes:

- <type> refers to the type of XML, in this case 'action', as opposed to 'initial' which is the first type of request.
- <name> is the name of the action.
- field <name> is the name of the field.
- field <value> in this instance is the menu selection made by the user.

4.1.4. Action response XML <ussdAppResponse>

Below is a sample of a response XML file that the client application will need to send back to USSD in response to a USSD request (after the first one):

```
<ussdAppResponse>
  <prompt>27731234567</prompt>
  <state>end</state>
</ussdAppResponse>
```

Note: <state> indicates to USSD where in the definition XML it needs to go next.

- '' - USSD needs to continue to pass strings of data between the client application and the mobile phone
- 'end' - the session should be ended

- ‘?’ - the next state is defined by the <nextState> tag in the definition XML
- ‘incorrect’ – the next state is the previous state (user to re-enter/ correct input entered in the previous transaction)

Important: These are examples only. The tags in the XML files will vary depending on the requirements of individual applications.

4.1.5. Send aborts request XML <ussdAppRequest>

Below is a sample of a request XML file that will be sent to the client application indicating the user has either terminated the session or the session has timed out. The client application does NOT have to respond to this request.

```
<USSDAppRequest>
  <type>abort</type >
  <msisdn>27731234567</msisdn>
  <sessionId>721082474</sessionId>
  <reason>TIMEOUT</reason>
  <network>MTN</network>
</USSDAppRequest>
```

Notes:

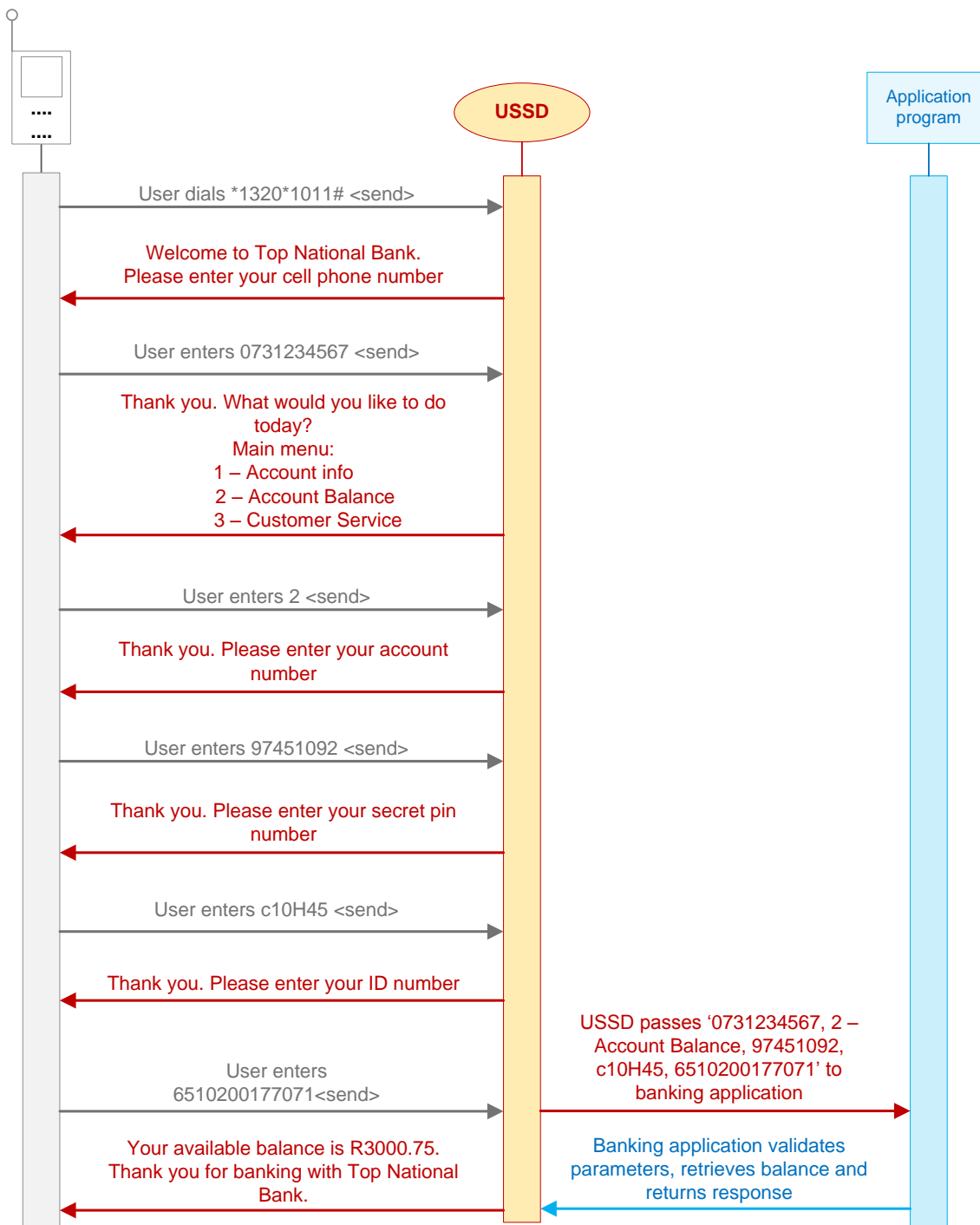
- <type> refers to the type of request this is, in this case ‘abort’. As we saw earlier, other types are ‘initial’ (which is the first type of request) or ‘action’ (which specifies subsequent requests).
- <reason> can be ‘timeout’ or ‘abort’ (where a user terminates session before it is complete).
- An Affiliate does NOT need to respond to this request.

5. Appendix A – additional examples

Note: In the examples below we are using USSD to format some of the prompts to the user. We could also use USSD purely as a portal in which case the client application would format all the prompts. In that scenario, USSD would simply transmit strings of data between the user and the client application.

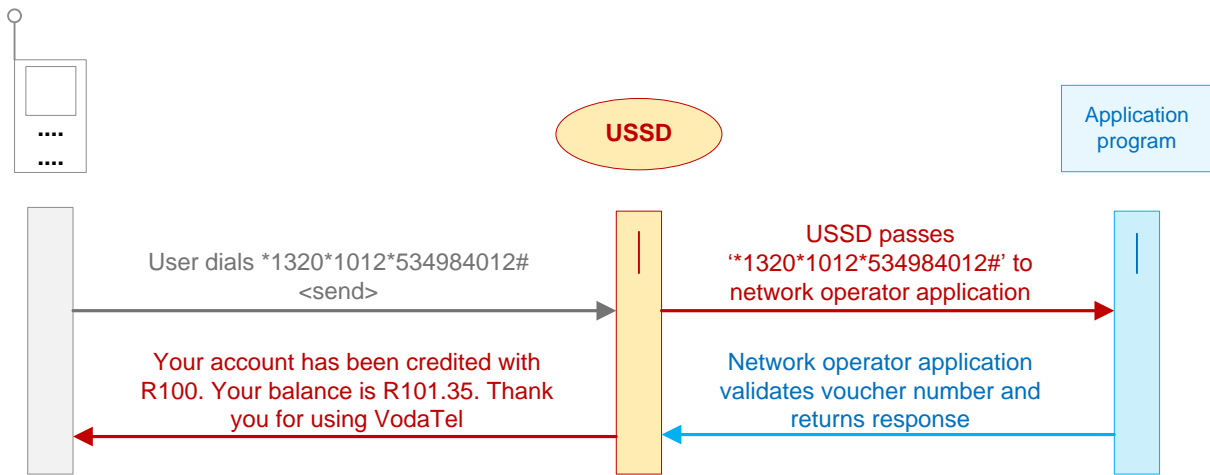
5.1. Bank account balance request

In this example, a user requests his bank account balance via Grapevine’s USSD. The menus presented by the USSD application are developed by Grapevine according to client specifications and allow the user to browse through various USSD services offered by the bank.



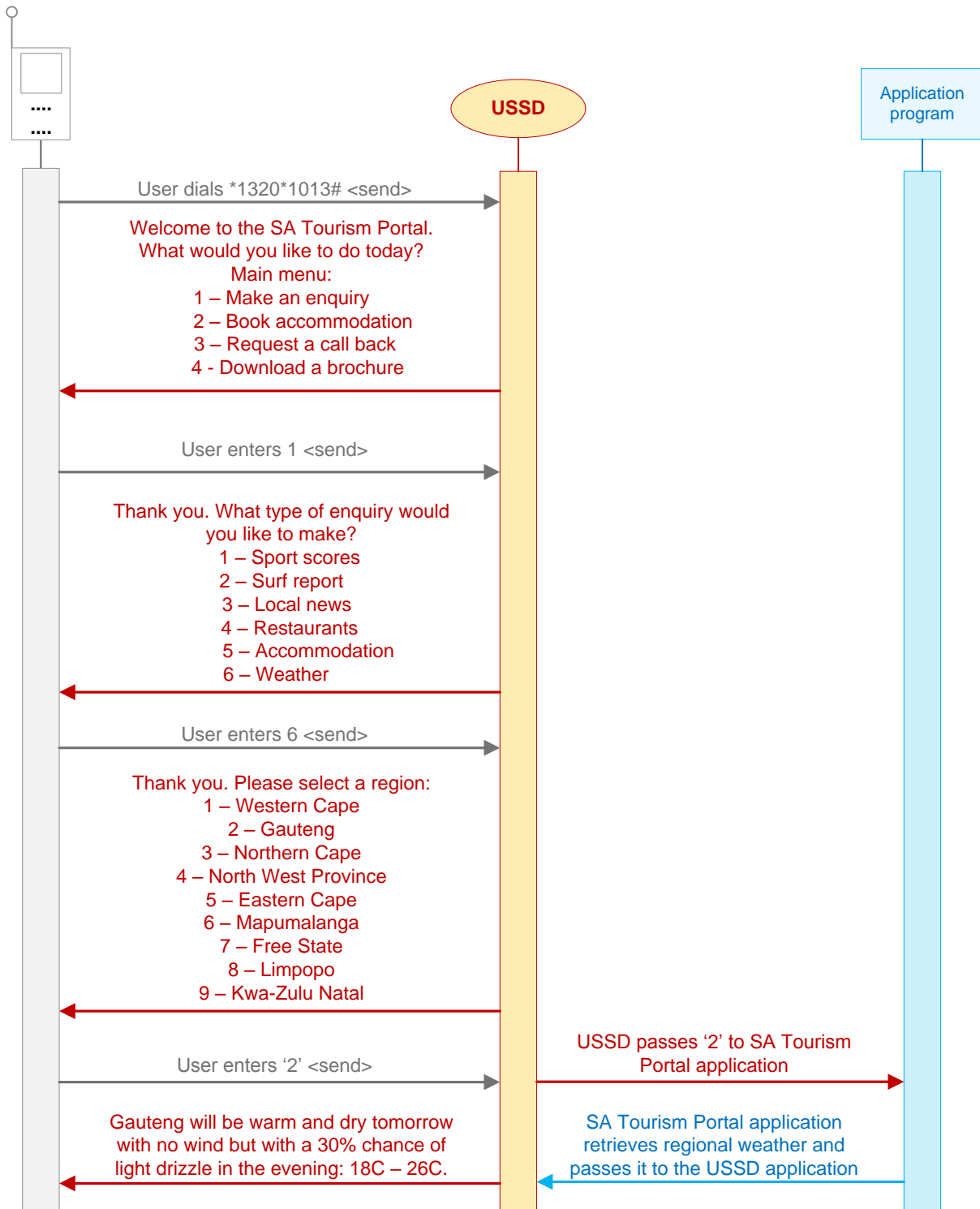
5.2. Top up pre-paid account balance request

In this example, a user enters a USSD string and voucher number to top up the prepaid account balance on his mobile phone. The request, including the voucher number parameter, is routed directly to the network operator and the user’s account balance is topped up immediately.



5.3. Regional weather request

In this example, Grapevine’s USSD allows a user to browse SA Tourism Portal USSD services. The user chooses to view the weather for his region. The menus for this service are developed by Grapevine according to client specifications.



5.4. Competition entry

In this example, a user enters the Sporting Heroes Competition. He sends a USSD string and the answer to the competition question directly to the WinALot company that is holding the competition. WinALot enters the user’s mobile phone number into the competition draw and sends a message to the user, telling him he will be notified if he wins the competition.

