1. **Introduction**
   1. **Purpose:**

The purpose of this document is to describe the external and processing requirements for the mobile billing application to be developed. The program implements the concept of message passing.

* 1. **Project Scope:**
* Call rates are assumed to be uniform.
* PULSE duration is assumed to be per second pulse
  1. **Definitions,Acronyms,Abreviations:**

**MPI:** Message Passing Interface

**IPC:** Inter Process Communication

**Message passing** in [computer science](http://en.wikipedia.org/wiki/Computer_science), is a form of communication used in [parallel computing](http://en.wikipedia.org/wiki/Parallel_computing), [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming), and [inter-process communication](http://en.wikipedia.org/wiki/Interprocess_communication). In this model [processes](http://en.wikipedia.org/wiki/Process_%28computing%29) or [objects](http://en.wikipedia.org/wiki/Objects) can send and receive messages (comprising zero or more bytes, complex data structures, or even segments of code) to other processes. By waiting for messages, processes can also [synchronize](http://en.wikipedia.org/wiki/Synchronization_%28computer_science%29).

* 1. **References:**

[**www.wikipedia.com/messagepassing/main.php**](http://www.wikipedia.com/messagepassing/main.php)

[**www.unixtalks.com/index.html**](http://www.unixtalks.com/index.html)

[**www.mpiprogramming.com/1.php**](http://www.mpiprogramming.com/1.php)

1. **General Description**
   1. **Product Function Overview:**

Communication between processes is necessary for proper access and manipulation of resources. Message passing provides a method for inter process communication(IPC). Shared memory method, also used for IPC posed many disadvantages. Hence message passing techniques have evolved.

* 1. **Product Functions**

The product makes use of an interface providing users to choose between their connection typei.e, postpaid or prepaid. Either of the selection provides the user with an interface to choose between making a call and generating a bill.

Both the type of connections are dealt with separately and are given different types of processing.

* 1. **User Classes and characteristics**

The User for this system is anyone who requests for a call or generation of bills. Rather than characterizing the user, the calls made by them can be characterized as prepaid or postpaid type.

For the postpaid type, the bill amount is incremented as per the calls made.

For the prepaid type, the account balance is decremented as per the calls made.

* 1. **Operating Environment**

The mobile billing system and the message passing method used is implemented on Linux platform.

1. **External Interface Requirements**
   1. **User Interfaces**

The User interface provides the facility to choose between connection types(prepaid or postpaid). It also allows the user to either make a call or request a bill.

* 1. **Communication Interfaces**

This system uses an MPI to communicate between the process which request an operation and the process dedicated for computations.

Also a synchronization mechanism (using semaphores) is used to provide access privileges to requesting processes.

1. **System Features**
   1. **Description and Priority**

The synchronization between various processes have to be given a higher priority to prevent data inconsistency and efficient processing of user requests.

* 1. **Stimulus/Response Sequences**

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| ***Prepaid connection type*** | ***Access file related to prepaid users*** |
| ***Postpaid connection type*** | ***Access file related to postpaid type*** |
| ***Call Request*** | ***Start call timer*** |
| ***Call Terminate request*** | ***End call Timer and update the corresponding file*** |
| ***Balance Enquiry*** | ***Send balance/bill*** |

* 1. **Functional Requirements:**

REQ-1: The system has to be able to access the corresponding files.

REQ-2: Multiple requests should be handled.

REQ-3: User process and the process involved in handling requests have to communicate using a suitable interface.

REQ-4: Processes for handling the requests have to handle various computations and updating.

1. **Non Functional Requirements**
   1. **Security Requirement:**

The user information maintained by the service provider cannot be mishandled as no user can access any other user data and thus security of data is implemented.

* 1. **Performance Requirement:**

There is no particular performance requirement but to run the application in an efficient way the system should be free of errors and correct inputs must be given .

* 1. **Availability:**

There should be a good number of users requesting for any processing activity so that some message passing can be done to lead efficient synchronization as well as inter-process communication.

* 1. **Portability:**

The application will be designed on a platform with few packages installed.Thus the system to be developed would not be portable one.

* 1. **Maintainability:**

The application will require very less maintenance on the developers part as the functionalities; whichever are specified are implemented completely.

Future development of the code will be done by the developers on regular bases.

Errors should be removed as and when they are found which may lead to a wrong output.