Ministry of Communication and Information Technology
Information Technology Institute
Professional Training Program – Intake 41

CONVERGENT BILLING SYSTEM
PROJECT TECHNICAL MANUAL

Prepared By

Aya Abdelkader Abdelsalam
Aya Magdy Hamouda
Hala Elmetwally Ramadan
Mayada Tarek Mohamed

Supervised By

ENG: Mohamed Elsabagh

INFORMATION TECHNOLOGY INSTITUTE
9-MONTH PROGRAM – INTAKE 41
GRADUATION PROJECT

Project Code	INT41-SV-Tel-01
Date:	30 June 2021
Revision:	Version 1.0

Table of Contents

ACKNOWLEDGEMENT

First of All, we would like to express our sincere gratitude to our supervisor Eng. Mohamed Elsabagh for his continuous support to our thesis, study and research, for his patience, motivation, enthusiasm, immense knowledge and also for his assistance in the technical aspects needed to enhance our project, his guidance helped us in all the time of this project, we would like to express our deep appreciation to all those who helped us to complete this project.

ABSTRACT

The telecommunications industry is one of the most developed industries, as many new services always appear Adapted to the evolution of technology. With this rapid development and companies seeking to develop their systems to be able to serve this huge number of people and deal with countless services, each service has its own characteristics and a method for calculating its cost. The challenge here is how to provide services with the highest quality and achieve the desired profit together taking into account the necessary of speed as a result of the large number of users and the nature of the services that most of them are at the real time. The billing system is one of the most important systems in telecommunications, because it is responsible for calculating consumption of users and therefore must be very accurate and fast to serve the largest possible number of users.

CHAPTER 1
INTRODUCTION

1.1 Project Description

This project intends to implement an open source billing system in a telecommunication network that collect consumed data of voice, SMS and internet usages, calculate charging and billing information, produce bills at specific bill cycles to customers, and process their payments.

The motivation for this project is to introduce an open source platform that is as good as the billing of the vendors, with high availability and scalability.

1.2 Business Objectives and Success Criteria

The main objectives of this project are:

- Create an effective and accurate billing system for Telecommunication companies to be able to guarantee their revenue.
- This billing system is based on recording customer, transactions in order, collecting usage, aggregating it, applying required charges, record and store payments made to customer accounts and finally generating invoices for the customers.
- The success of this project can be measured by achieving an efficient working billing system and eligibility for implementation and utilization in a network operator.

1.3 Scope

The scope of the project is to build a Billing system in order to issue an invoice to the customer every month by collecting different CDRs (which describe the use service, and used to rate the services) in different data formats like BINARY, then converts it into a normalized output format using ASN, then the Rating Engine receives the events in the form of data records(CDRs) to calculate the cost of the services as per the rate plan and apply any discounts, then it stores the rated services in the database for a billing purpose.

1.4 Constraints

Dimension	*Constraint* *(state limits)*	*Driver* *(state objective)*	*Degree of Freedom* *(state allowable range)*
Features	Genaric and have friendly GUI	To meet the success criteria. Evolution plan of the project	70-80% of high priority features included in release 1.0
Cost	N/A	N/A	N/A
Schedule	Project has to be completed 4 days at min before the defense date	To allow enough time testing, documentation and enhancements	Not less than 7 days
Staff	maximum team size is 4 developers	This team size is appropriate to accomplish the planned requirements and to gain max benefits and experience	N/A

1.5 Vision

Our vision is to achieve a scalable, effective, reliable, accurate, secure, open and highly available system. This system allows operators to register data of customers and modify or add different services to customers, also allow the customer to use the system to display invoices or download it in a secure manner.

CHAPTER 2
PROJECT SPECIFICATIONS

2.1 Billing System Overview

2.1.1 Objectives

- To understand the Billing Work file.
- To understand the origination of costs for time.
- To create, print, and void invoices.
- To create and record accounting journal entries.

2.1.2 About the Billing Process

- The typical billing process includes accumulating costs, creating and printing invoices.
- When you perform the billing process using the Contract Billing system, you can:
  - Accumulate billable costs from multiple systems.
  - Calculate the billing amounts at the time you create invoices.
  - Facilitate immediate billing upon entering the costs into the system.
  - Print invoices to customer specifications.
  - Automatically create and record the journal entries for the Accounts.
  - Receivable and General Accounting systems that result from billing.

2.1.3 Billing Process Tasks

- Accumulating costs.
- Working with the customer file history.
- Generating invoices automatically.
- Working with invoices.
- Creating invoices manually.
- Printing invoices.
- Working with final invoices

2.2 Mediation Overview

2.2.1 Mediation platform

They are often referred to as internet protocol detail records. Mediation also processes event detail records or EDRs that are usually generated by the charging systems or any other network elements. Hence, Mediation platforms can process all xDRs that are generated by any network element, whether if it is CDR, EDR or UDR.

2.2.2 Data Type

- The usage and call detail record data types hold data such as NPX, NPA, call duration, peak time flag and call length. Data may be represented in ASCII or binary formats. The billing mediation platform typically reads this data and converts it into common normalized format.
- Billing systems and all other downstream systems, in turn, converts this data to component [its own] understandable format.

2.2.3 Functionality

Typically a mediation platform is used for the following tasks:

- Collection and validation of call detail records
- Filtering out of non-billing-relevant call detail records
- Collating
- Correlation of different input sources call detail records
- Aggregation of partial call detail records related to the same call
- Format change and call detail record normalization
- Business transformation of data

In a telecom billing scenario, mediation is the first step after receiving a call detail record.
The mediated call detail record files are forwarded to a rating engine, which calculates the charge associated with the call detail records. In today's world, Rating Engines are becoming more necessary for the telecom billing system to be able to meet the growing variant customer needs for different services.

Sophisticated billing mediation software serves end to end functionality for telecom operators. Mediation software performs various operations from data collection to downstream distribution to modules like retail billing, roaming, interconnect settlement, business intelligence, fraud detection and revenue assurance.

2.2.4 Supported systems

Telecom operators offer voice, video, data, fax and internet services to subscribers and partners on various product lines. Mediation products are tuned to provide solutions for complex business challenges. The call data is produced by network devices in the form of call detail record.

Mediation platforms for telecom practice support various systems:

- Retail Billing.
- Wholesale Billing - National and International.
- Network Traffic Management Tools.
- Data Warehousing Systems.
- Business Intelligence / Big Data Systems.
- Reconciliation Systems.
- Fraud Management Systems.
- Provisioning feed to sub-systems.
- RA - Revenue Assurance Systems.
- ICT - Information and Communication Technology Systems.
- Assurance Insight Synap System.

2.3 Call Detail Record (CDR)

CDR is a data record produced by a telephone exchange or other telecommunications equipment that documents the details of a telephone call or other telecommunications transaction (e.g., text message) that passes through that facility or device. The record contains various attributes of the call, such as time, duration, completion status, source number, and destination number. It is the automated equivalent of the paper toll tickets that were written and timed by operators for long-distance calls in a manual telephone exchange.

A call detail record contains data fields that describe a specific instance of a telecommunication transaction, but does not include the content of that transaction. By way of simplistic example, a call detail record describing a particular phone call might include the phone numbers of both the calling and receiving parties, the start time, and duration of that call. In actual modern practice, call detail records are much more detailed, and contain attributes such as:

- The phone number of the subscriber originating the call (calling party, A-party).
- The phone number receiving the call (called party, B-party).
- The starting time of the call (date and time).
- The call duration.
- The billing phone number that is charged for the call.
- The identification of the telephone exchange or equipment writing the record.

- A unique sequence number identifying the record.
- Additional digits on the called number used to route or charge the call.

- The disposition or the results of the call, indicating, for example, whether or not the call was connected.
- The route by which the call entered the exchange.
- The route by which the call left the exchange.
- Call type (voice, SMS, etc.).
- Any fault condition encountered.

2.4 Abstract Syntax Notation number One (ASN.1)

2.4.1 Main Concepts

ASN.1 is a formal notation used for describing data transmitted by telecommunications protocols, regardless of language implementation and physical representation of these data, whatever the application, whether complex or very simple.
In telecommunications and computer networking, ASN.1 is a standard and flexible notation that describes data structures for representing, encoding,
transmitting, and decoding data. It provides a set of formal rules for describing the structure of objects that are independent of machine-specific encoding techniques and is a precise, formal notation that removes ambiguities.

The notation provides a certain number of pre-defined basic types such as:

- Integers (INTEGER)
- Booleans (BOOLEAN)
- Character strings (IA5String, UniversalString...)
- Bit strings (BIT STRING)

ASN.1 defines the abstract syntax of information but does not restrict the way the information is
encoded. Various ASN.1 encoding rules provide the transfer syntax (a concrete representation)
of the data values whose abstract syntax is described in ASN.1.
Some of the standard ASN.1 encoding rules are:

- Basic Encoding Rules (BER)
- Canonical Encoding Rules (CER)
- Distinguished Encoding Rules (DER)

ASN.1 together with specific ASN.1 encoding rules facilitates the exchange of

structured data especially between application programs over networks by describing data structures in a way that is independent of machine architecture and implementation language.
In our project we were able to decode CDRs with ASN.1 schema using BinaryNotes tool.

2.4.2 BinaryNotes

BinaryNotes is the Open Source ASN.1 (Abstract Syntax Notation One) framework for Java and .NET.

The framework contains:

- Encoding/decoding library. The library has BER (Basic Encoding Rules), PER (Packet Encoding Rules) and DER (experimental) implementation.
- BNCompiler - the extensible (based on XSL) ASN.1 compiler which is able to generate the simple Java classes for the specified ASN.1 input file. The generated code has annotations/attributes that uses the compiler in runtime. You can customize the generated files by change the original XSL-templates or create your own templates.

BinaryNotes Compiler (BNCompiler) will generate the class/method declarations for The compiler generates ready to use code with the BinaryNotes library.
The generated classes has annotations/attributes and simple properties which uses by runtime library.

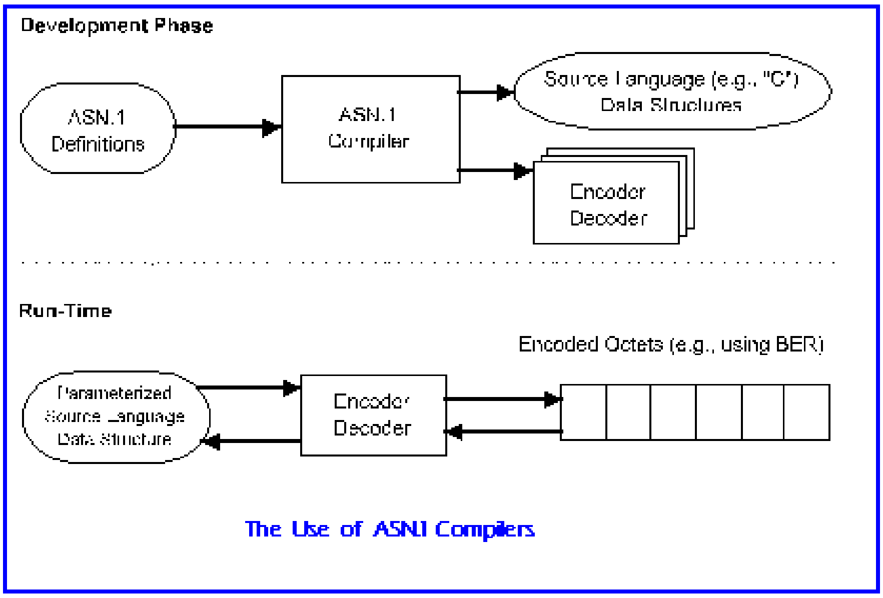

Development Phase
ASN.1 Definitions
ASN.1 Compiler
Source Language (e.g., "C") Data Structures
Encoder Decoder
Run-Time
Encoded Octets (e.g., using BER)
Parameterized Source Language Data Structure
Encoder Decoder
The Use of ASN.1 Compilers

Figure 2.4.1 ASN.1 Decoding And Encoding

2.5 Jasper Reports

A report is a meaningful, well-defined, and summarized presentation of information. Usually, the routine activities are automated and data summarized into a decision-supporting "Reports". Reports represent usual messy data into charts, graphs, and other forms of graphical representations.

JasperReports was started by Teodor Danciu, in 2001, to provide reporting capabilities to the project he was working on. Later that project got canceled; but he continued working on JasperReports in his spare time. Since its first release back in November 2001, JasperReports has become immensely popular, and is currently one of the most popular Java reporting tools available. It is an open source Java reporting tool that can write to a variety of targets, such as screen, a printer, into PDF, HTML, Microsoft Excel, RTF, ODT, comma-separated values (CSV) or XML files. It can be used in Java-enabled applications, including Java EE or web applications, to generate dynamic content.

- Features of JasperReports

Some of the significant features of JasperReports are It has a flexible report layout, can present data either textually or graphically, Developers can supply data in multiple ways, can accept data from the multiple data sources, can generate watermarks (A watermark is like a secondary image that is laid over the primary image), can generate sub reports and It is capable of exporting reports in a variety of formats.

- Jasper Managers Classes

JasperReports, like most reporting applications, uses report templates structured in multiple sections, such as title, summary, detail, and page and group headers and footers. Each section has a free-form layout in which you can place various types of elements, including images, static and dynamic text fields, lines, and rectangles. The reporting engine uses this template to organize data in an XML file (JRXML) or to create it programmatically using the library’s API. This data may come from various data sources, including relational databases, collections, or arrays of Java objects or XML data. Users can plug the reporting library into custom-made data sources by implementing a simple interface. There are number of classes, which will be used to compile a JRXML report design, to fill a report, to print a report, to export to PDF, HTML & XML files, view the generated reports, and report design.

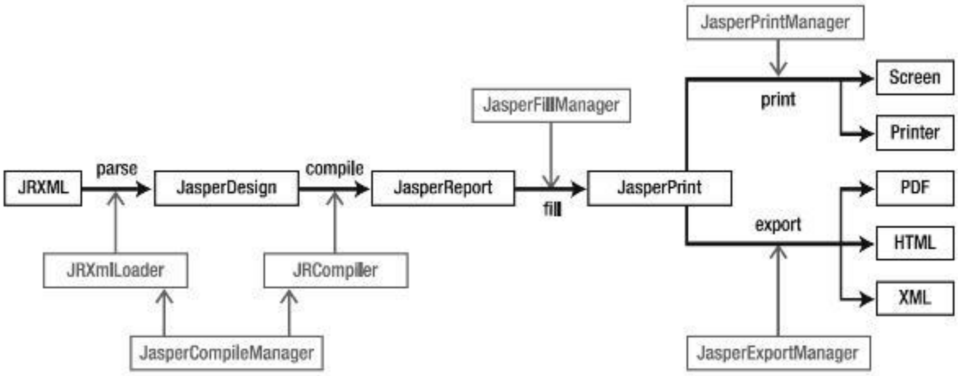


Figure 2.5.1 JasoerReports Working Flow

The list of these classes is:

- net.sf.jasperreports.engine.JasperCompileManager is used to compile a JRXML report template.
- net.sf.jasperreports.engine.JasperFillManager is used to fill a report with data from the data source.
- net.sf.jasperreports.engine.JasperPrintManager is used to print the documents generated by the JasperReports library.
- net.sf.jasperreports.engine.JasperExportManager is used to obtain PDF, HTML, or XML content for the documents produced by the report-filling process.
- net.sf.jasperreports.view.JasperViewer is used to represent a simple Java Swing application, which can load and display reports.
- net.sf.jasperreports.view.JasperDesignViewer is used at design time to preview the report templates.

CHAPTER 3
PROJECT STRUCTURE

3.1 Billing Database

3.1.1 Entity Relationship Diagram(ERD) Introduction

graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. An ERD uses data modeling techniques that can help define business processes and serve as the foundation for a relational database.

The importance of ERD sis that it provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a reference point, should any debugging or business process re-engineering be needed later.

However, while an ERD can be useful for organizing data that can be represented by a relational structure, it can't sufficiently represent semi-structured or unstructured data. It's also unlikely to be helpful on its own in integrating data into a pre-existing information system.

3.1.2 Creation of ERD

graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system. An ERD uses data modeling techniques that can help define business processes and serve as the foundation for a relational database.

ERDs are generally depicted in one or more of the following models:

- A conceptual data model, which lacks specific detail but provides an overview of the scope of the project and how data sets relate to one another.

- A logical data model, which is more detailed than a conceptual data model, illustrating specific attributes and relationships among data points. While a conceptual data model does not need to be designed before a logical data model, a physical data model is based on a logical data model.

- A physical data model, which provides the blueprint for a physical manifestation such as a relational database of the logical data model. One or more physical data models can be developed based on a logical data model.

There are five basic components of an entity relationship diagram. Similar components will be designated by the same shape. For example, all entities types might be enclosed in a rectangle, while all attributes are enclosed in a diamond. The components include:

1. Entities, which are objects or concepts that can have data stored about them. Entities refer to tables used in databases.
2. Attributes, which are properties or characteristics of entities. An ERD attribute can be denoted as a primary key, which identifies a unique attribute, or a foreign key, which can be assigned to multiple attributes.
3. The relationships between and among those entities.
4. Actions, which describe how entities share information in the database.
5. Connecting lines.

3.1.3 Database Structure

There are several entities in the project: Customer, Profile, Service, CDR, Rating, Free Units and Invoice.

- Customer

Each Customer has a customer id which is unique, email which must include @ sign, name, MSISDN which is separated for each customer, IMSI, Address and Profile Id which indicates the profile the customer signed in. Figure 3.1

The normal customer can log in website to see or download his invoices.

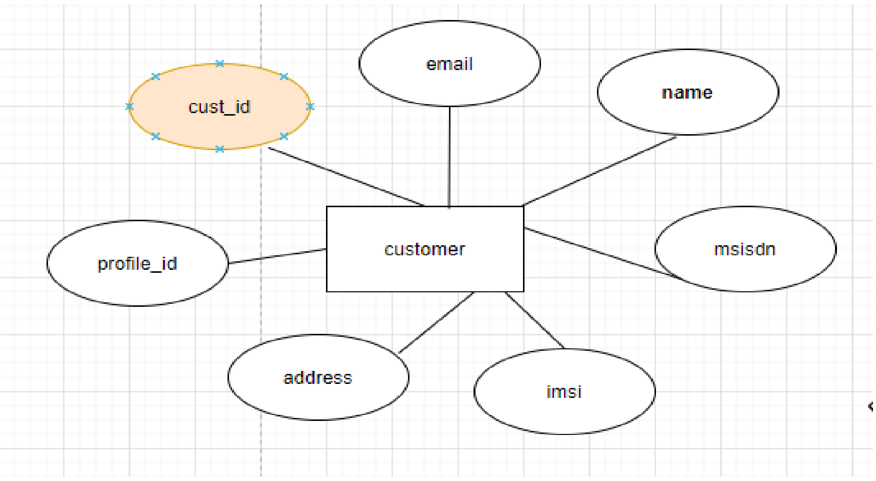


Figure 3.1.1 Customer Entity

- Profile

Each profile has a profile id which is unique, name and renew day . Figure 3.2

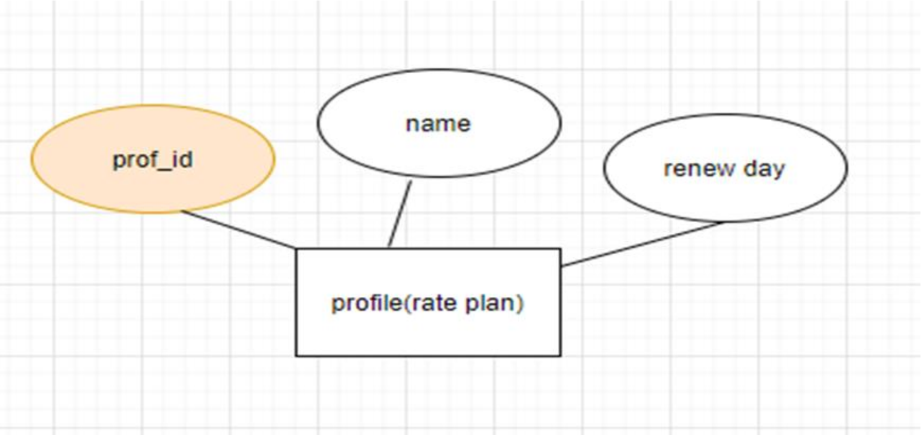


Figure 3.1.2 Profile Entity

- Service

Each service has a service id which is unique and name; each profile must have at least 3 main services; call, SMS and data. Figure 3.3

Admin can add new services with its fees.

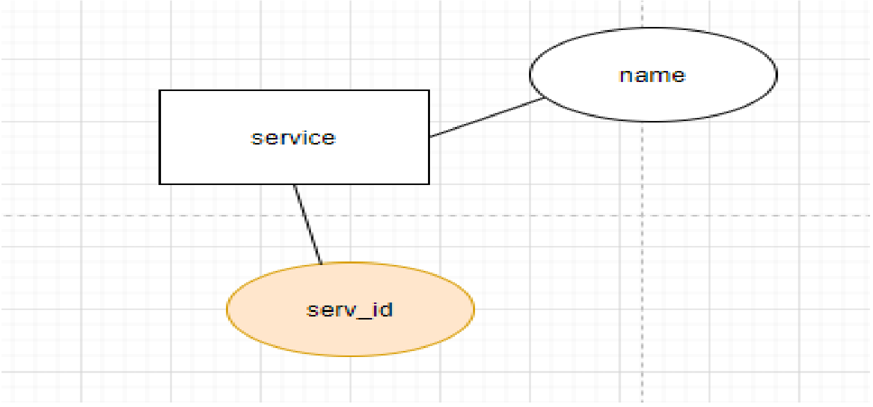


Figure 3.1.3 Service Entity

- CDR

Phone number of the subscriber originating the call (calling number).
Phone number receiving the call (called number).
Starting time of the call (date and time).
Call duration.
Call type (voice, SMS, data, etc.).
Consumption.
Date.
Figure 3.4

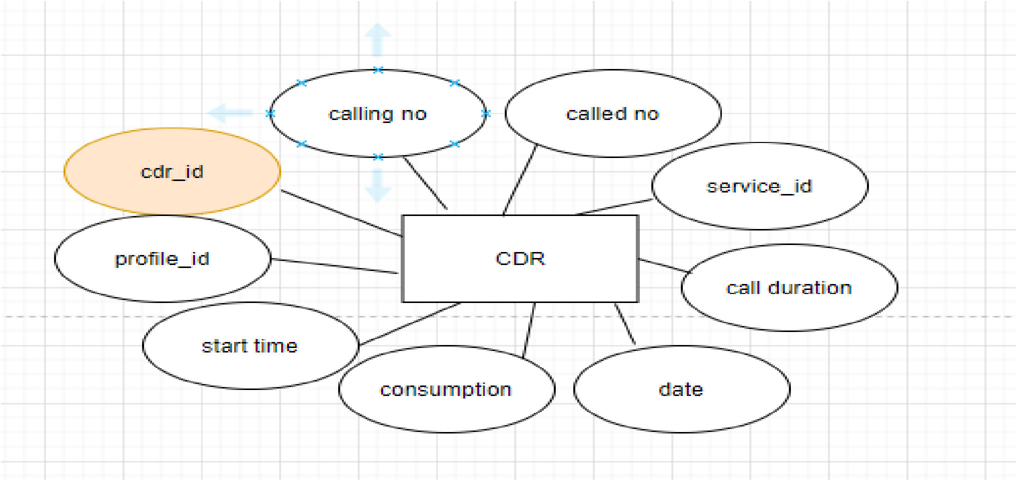


Figure 3.1.4 Service Entity

- Free Units

Each customer has free units which can be call units, SMS units or data units on net or cross net based on the profile (rate plan) which he subscribed. Figure 3.5
The unique key is free_id.

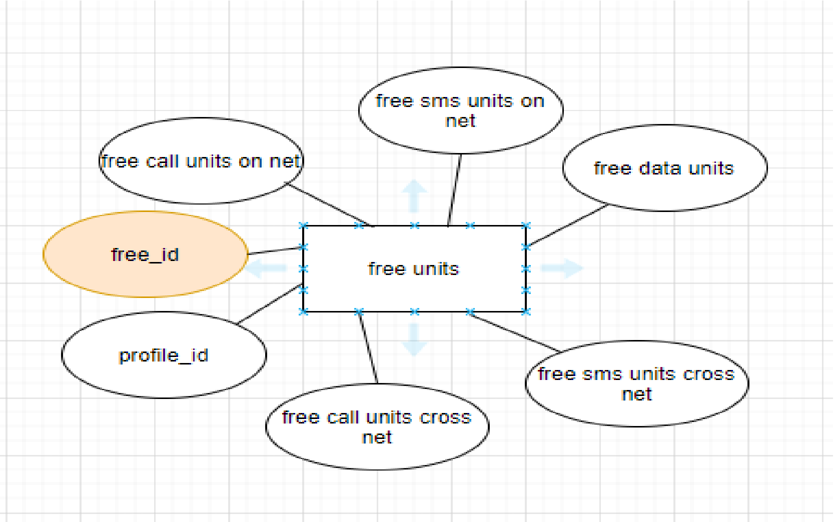


Figure 3.1.5 Profile Entity

- Rating

Based on the profile (rate plan) and according to the service we specify fees if the service is on net, fees cross net, fees international, quota to the same operator, quota to other operators and quota international. Figure 3.6

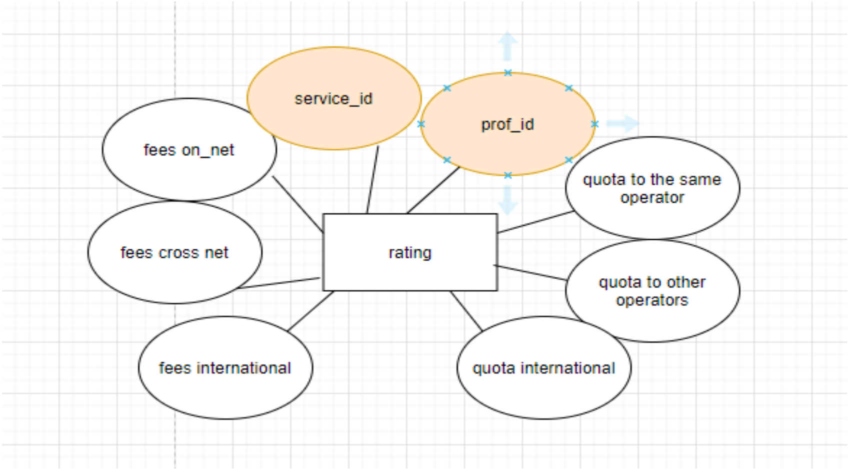


Figure 3.1.6 Rating Entity

- Invoice

Every month there is an invoice which is generated for each customer including his usage of services during month.

It includes consumption units in voice, SMS data, total fees and his name which is brought by his id from customer table. Figure 3.7

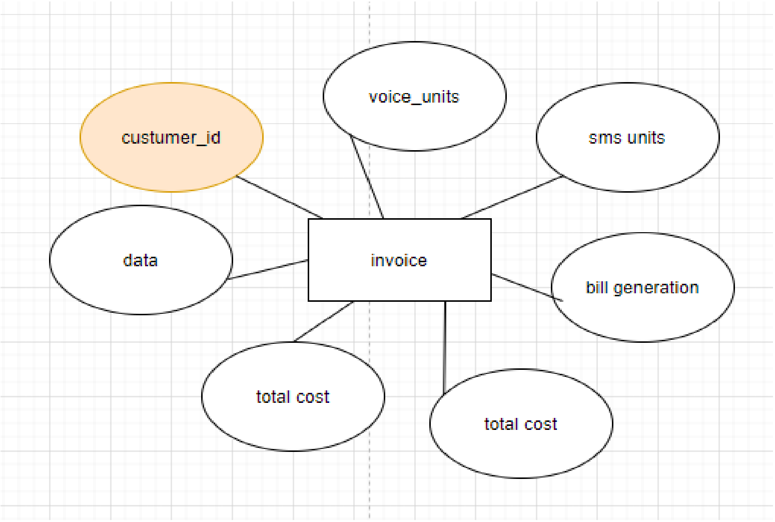


Figure 3.1.7 Invoice Entity

- Relations Between Entities

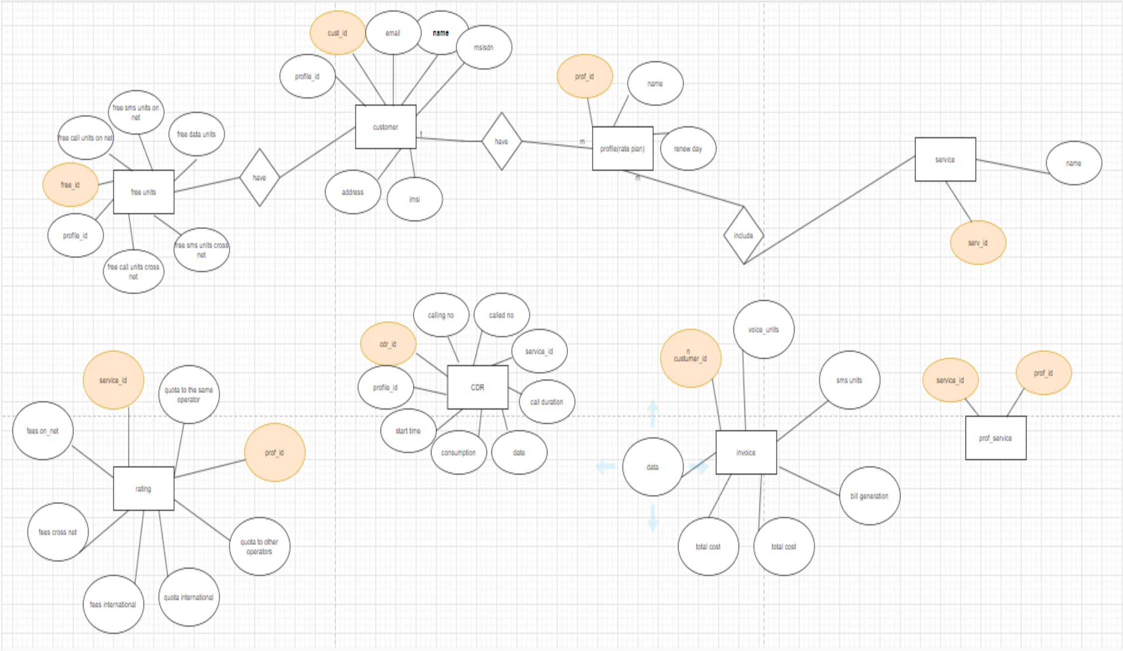


Figure 3.1.8 Relations between Entities

3.2 Billing Portal

3.2.1 For Normal User

For both of normal user and Admin you have to login to your page through your user name and password as shown in figure 3.2.1.
After logging in, you can see your invoices for each month as a PDF file and you can download them easily.

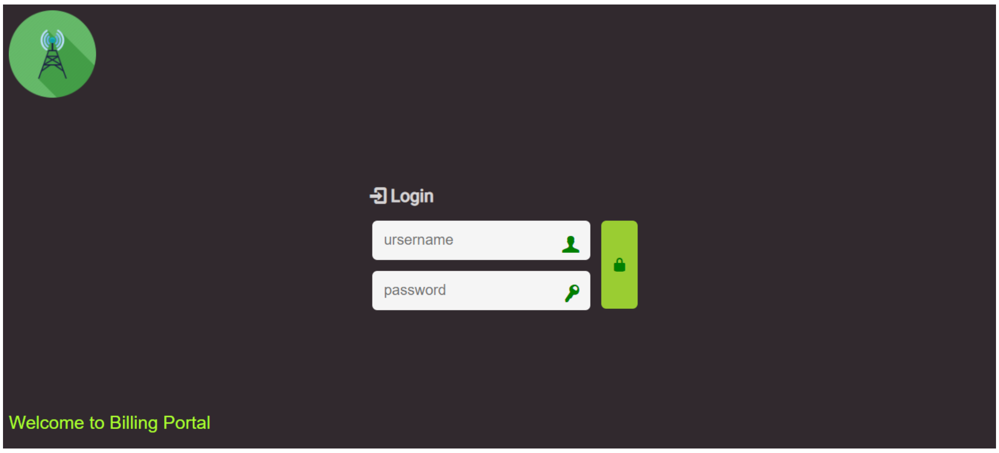


Figure 3.2.1 Logging in page

3.2.2 For Admin

You have the ability to add new customers with free units, add new profiles with fees for each service, display all customers, profiles and invoices and to edit anyone of them. Figure 3.2.2.

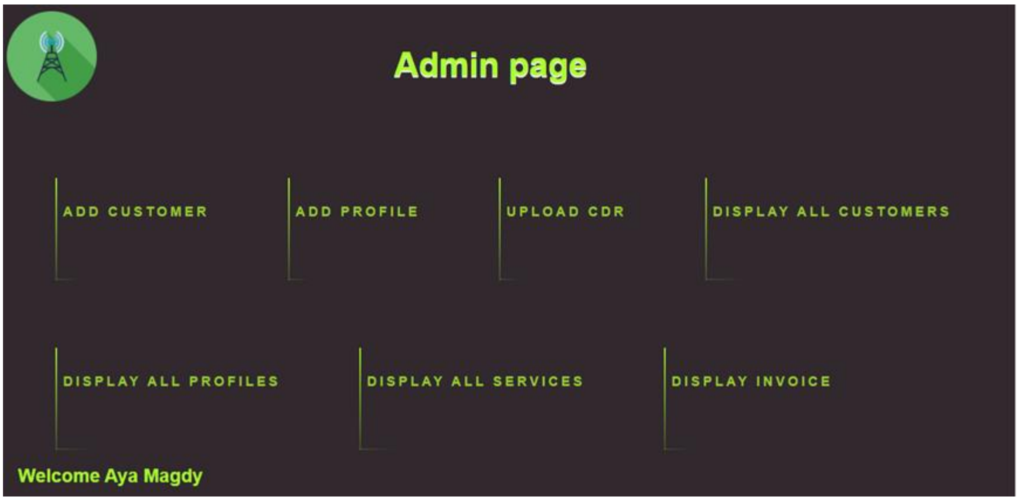
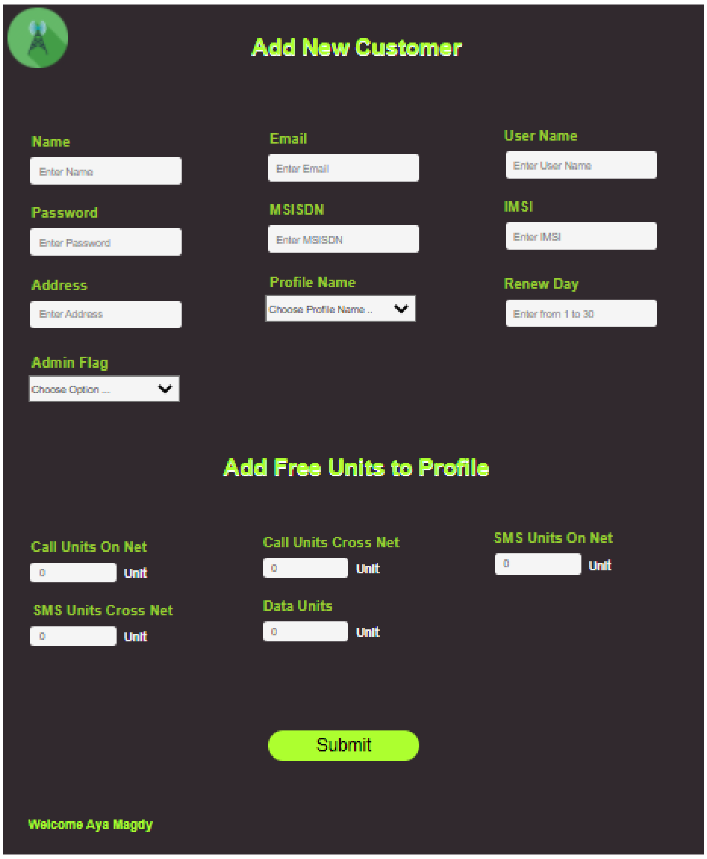


Figure 3.2.2 Admin page

- Add Customer



Add New Customer

Name: Enter Name

Email: Enter Email

User Name: Enter User Name

Password: Enter Password

MSISDN: Enter MSISDN

IMSI: Enter IMSI

Address: Enter Address

Profile Name: Choose Profile Name ..

Renew Day: Enter from 1 to 30

Admin Flag: Choose Option ...

Add Free Units to Profile

Call Units On Net: 0 Unit

Call Units Cross Net: 0 Unit

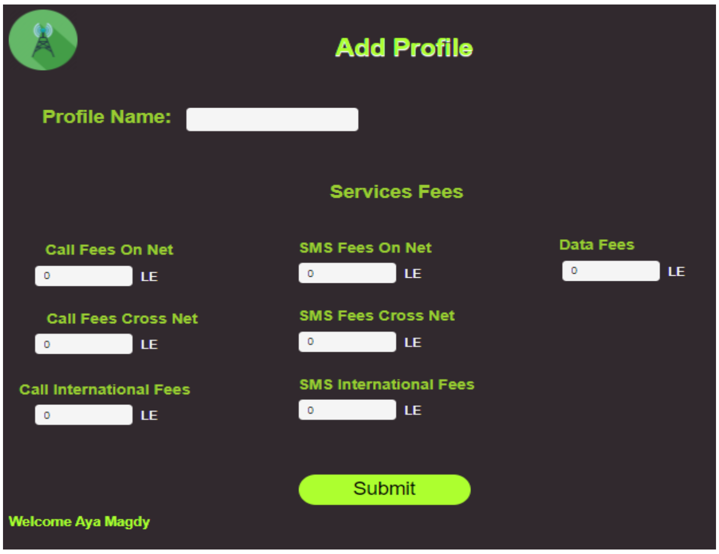
SMS Units On Net: 0 Unit

SMS Units Cross Net: 0 Unit

Data Units: 0 Unit

Figure 3.2.3 Add Customer Page

- Adding Profile



Add Profile

Profile Name:

Services Fees

Call Fees On Net: 0 LE

SMS Fees On Net: 0 LE

Data Fees: 0 LE

Call Fees Cross Net: 0 LE

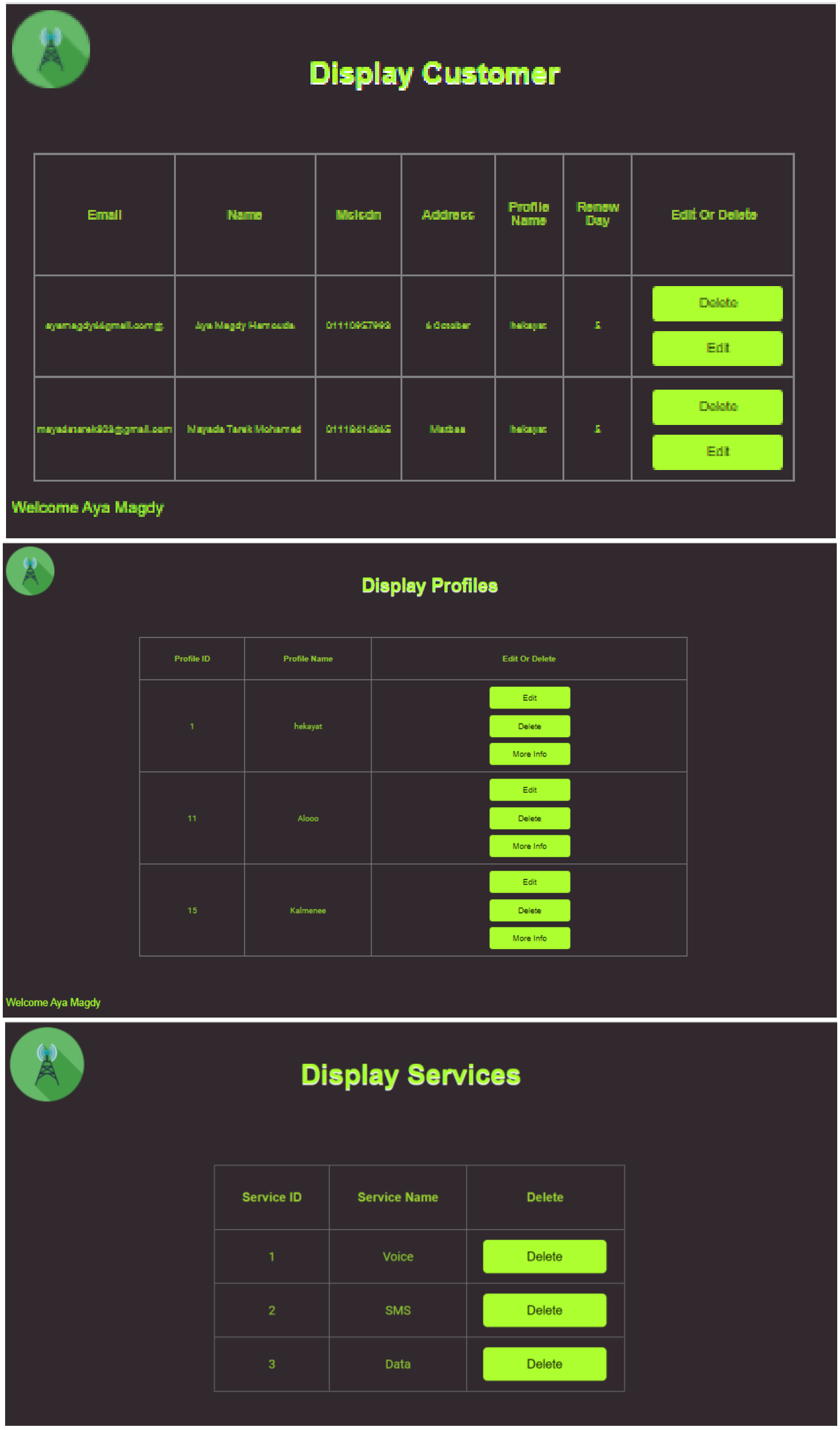
SMS Fees Cross Net: 0 LE

Call International Fees: 0 LE

SMS International Fees: 0 LE

Figure 3.2.4 Add Profile Page

- Display Customers, Profiles And Services



Display Customer

Email	Name	Msisdn	Address	Profile Name	Renew Day	Edit Or Delete
ayamagdy44@gmail.com@	Aya Magdy Hamouda	01110957993	6 October	hekayat	5	Delete Edit
mayadatarek903@gmail.com	Mayada Tarek Mohamed	01119814965	Masbas	hekayat	5	Delete Edit

Welcome Aya Magdy

Display Profiles

Profile ID	Profile Name	Edit Or Delete
1	hekayat	Edit Delete More Info
11	Alooo	Edit Delete More Info
15	Kalmenee	Edit Delete More Info

Welcome Aya Magdy

Display Services

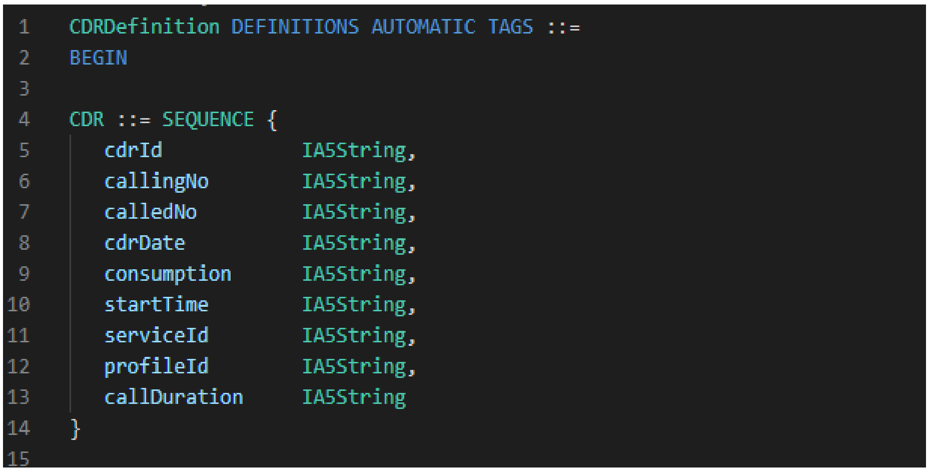
Service ID	Service Name	Delete
1	Voice	Delete
2	SMS	Delete
3	Data	Delete

Figure 3.2.5 Display Customers, Profiles and Services

3.3 CDR Parser Process

CDR parser is the process used in order to decode CDRs using BinaryNotes, we have to create ASN.1 declaration related to the encoded CDRs the portal will receive

For the following simple ASN.1 declaration:



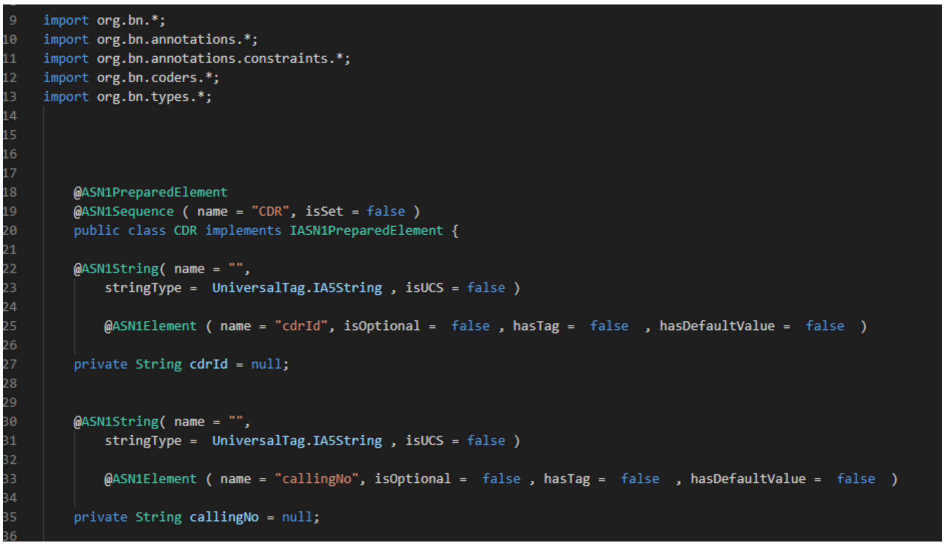
```
CDRDefinition DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

CDR ::= SEQUENCE {
    cdrId            IA5String,
    callingNo        IA5String,
    calledNo         IA5String,
    cdrDate          IA5String,
    consumption      IA5String,
    startTime        IA5String,
    serviceId        IA5String,
    profileId        IA5String,
    callDuration     IA5String
}

```

Figure 3.3.1 ASN.1 declaration

The generated class will be as following:



```
import org.bn.*;
import org.bn.annotations.*;
import org.bn.annotations.constraints.*;
import org.bn.coders.*;
import org.bn.types.*;




    @ASN1PreparedElement
    @ASN1Sequence ( name = "CDR", isSet = false )
    public class CDR implements IASN1PreparedElement {

    @ASN1String( name = "",
        stringType =  UniversalTag.IA5String , isUCS = false )

        @ASN1Element ( name = "cdrId", isOptional =  false , hasTag =  false  , hasDefaultValue =  false  )

    private String cdrId = null;


    @ASN1String( name = "",
        stringType =  UniversalTag.IA5String , isUCS = false )

        @ASN1Element ( name = "callingNo", isOptional =  false , hasTag =  false  , hasDefaultValue =  false  )

    private String callingNo = null;

```

Figure 3.3.1 ASN.1 declaration

The declaration contains the objects that will be used by BinaryRate Compiler to generate java class to Decode the received CDR.

After generating the CDR Parser Code on the CDR.BER file, the data will be extracted and easy to insert into project database.

3.4 Rating Engine Process

The mediation system processes all CDRs and converts them into a format compatible with the downstream systems, one of these downstream systems is billing system. These CDRs are entered on the rating engine, which collects all the necessary information to calculate the final cost of each CDR, depending on the customer number, the type of service used, the system subscribed to, and whether he has any offers, discounts, or free units, or not. This process is done permanently on any CDR that enters the system and all these costs are stored and collected until the monthly bill is generated. After carrying out the rating process, the flag (is_rated) is raised so that it is not withdrawn again. The steps of this process can be summarized as follows:

1) Determine this CDR belong to which of the users:
As mentioned previously, the CDR contains the user's chip number, which enables us to know everything about the user and withdraw all information using it through the data base.

2) Determine the type of service and its consumption:
For example, it may be a call, a SMS, or data in the case of a call or a SMS. We look at the called number, on which it is determined whether the service is On net, Cross net or International, and each of them has a different price.

3) Extracting the fees of this service through the data base, depending on the profile:
Each user has a profile that contains the fees of services, which vary according to their type, whether they are on net or cross net, and so on.

4) Rating the CDR based on previous information: by Knowing the prices and consumption, the final cost of CDR can easily be calculated.

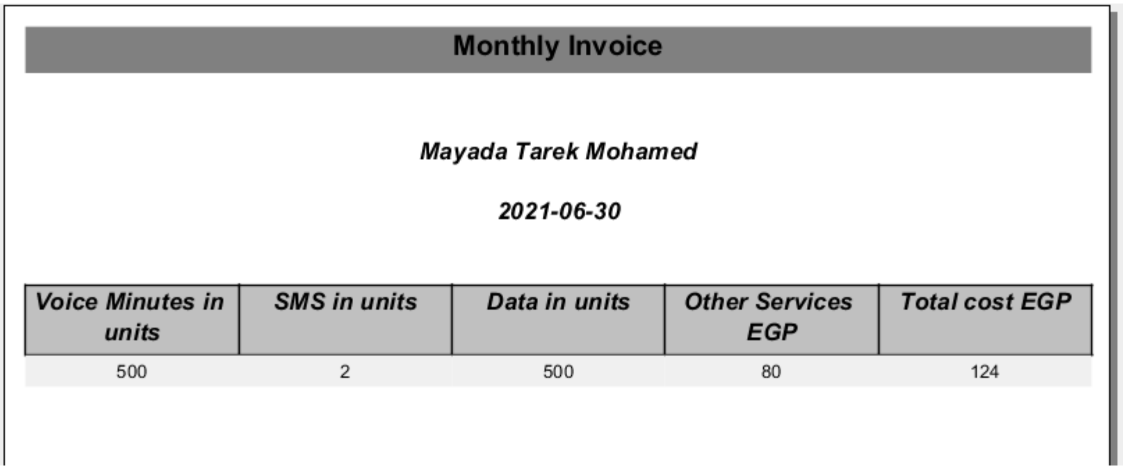
5) Check discounts or free units: This process is called re-rating in the event that the user has offers, discounts or units on this service that enables him to use it at a reduced or free form.

3.5 Bill Generator

After performing the rating process on each CDR that enters the system, other services performed by the user, such as subscribing to RBT (Ring Back Tone) or buying a headset or mobile in installments, are added to the bill. All these costs are added to the final bill for each service of this type. An OCC is generated for this user with its details and price, and it is stored, so that it is easy to add it at the end of the month on the bill. an invoice is generated for each user whose income is counted, the user’s consumption in all services and the final cost to be paid, depending on the monthly deadline for this user to pay the bill. Each user has

a specific day to pay the bill. This process is called bill generation and this is a Java Process is triggered to pull the information about all customers who agree to be repaid, which will be displayed in the invoice through the aforementioned consumer and invoice tables.

Using JasperReports, we can extract an invoice for each user, and as mentioned, the importance of JasperReports lies in the ability to extract dynamic reports and pull all the required information from the Database easily using Java Class, which creates a report for each user, which the user can then download from the portal. Each invoice contains the user's name, date, consumption of all services, and the final required cost, as shown in the figure 3.5.1.



Monthly Invoice

Mayada Tarek Mohamed

2021-06-30

Voice Minutes in units	***SMS in units***	***Data in units***	***Other Services EGP***	***Total cost EGP***
500	2	500	80	124

Figure 3.5.1 Final Bill Format

CHAPTER 4
USER GUIDE

4.1 Project Flow

This project aims to provide billing services to both postpaid users and admins using a web application connected to database server to provide these services.

As shown in Figure 4.1.1, web app services are:

For Admin

- Uploading CDRs:
  After uploading CDRs to the portal, the CDRParser java process will convert the CDR to understandable data that shows the usage consumption and can be saved in the customers' databases.
  After extracting the consumption data from CDR, the RatingEngine java process will calculate the cost of every user's consumption and save generated data in the rating database.
  BillGenerator java process is a process that generates bills for every user every month in the end of bill cycle.
- Create new customer.
- Edit customer data.
- Create new profile.
- Edit profiles.
- Display users.
- Display profiles.
- Display services.
- Display invoces.

For normal users

- Download invoices.

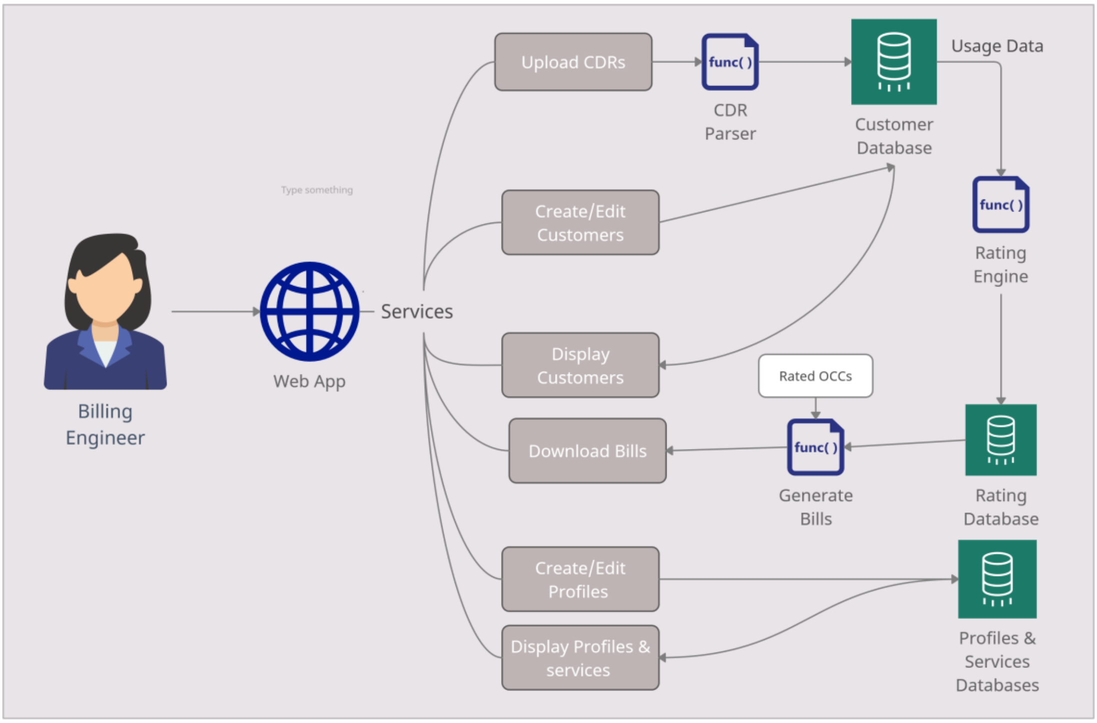


Figure 4.1.1 Web App Work Flow

4.2 Project Setup

Step 1: Download project zipped file and unzip it.
Step 2: Open the CMD and run the CDRParser jar to run CDR parser server.

```
java –jar File_Path/ CDRParser.jar
```

Step 3: Open another CMD window and run the RatingEngine jar to run rating server.

```
java –jar File_Path/RatingEngine.jar
```

Step 4: Restore the project attached database on your local server.
Step 3: Run the web app and start creating the application users.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

Mediation system is the gate to billing system which deliver CDRs to billing and rating system .through CDR rating system calculate fees for each service and send them to billing system to generate an invoice each month with total fees.

5.2 Future Work

Adding more options to the portal where the user can change his profile or the day of paying the bill easily through the web site. Each profile has a set of bundles that the customer can subscribe to, which are for a fixed fee every month with a number of units for each service depending on the type of bundle, which makes the bill specific and not absolute as in the previous situation. Adding terms to the consumption of the invoice according to the customer's desire.