A MAJOR-PROJECT REPORT

ON

"REMOTE DEPOSIT CAPTURE AND CASH APPLICATION AUTOMATION"

Submitted to

Kalinga Institute of Industrial Technology,

Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN COMPUTER ENGINEERING BY

RAKSHITA SINHA

ROLL NUMBER: 1605213

UNDER THE GUIDANCE OF

PROF ARUP SARKAR

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY,

Deemed to be University

BHUBANESWAR, ODISHA - 751024

2019-2020

School of Computer Engineering

Bhubaneswar, ODISHA 751024

CERTIFICATE

This is to certify that the project entitled

“REMOTE DEPOSIT CAPTURE AND CASH APPLICATION AUTOMATION”

submitted by

RAKSHITA SINHA

ROLL NUMBER: 1605213

is a record of bonafide work carried out by her, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science) at KIIT UNIVERSITY, Bhubaneswar. This work is done during year 2019-2020, under our guidance. The matter embodied in this project is original and has not been submitted for the award of any other degree.

Date: / /

(Sunil M. Nandam)	(Prof. Arup Sarkar)
Associate Director, Solution Delivery	Project Guide
HighRadius Corporation	

ACKNOWLEDGEMENT

I would like to take this opportunity to thank all my sources of aspiration during the course of the internship.

First and foremost, I express my deepest gratitude towards my mentor at KIIT Deemed to be University Bhubaneswar, **Prof. Arup Sarkar** for his valuable suggestions, insightful criticisms, and directions throughout. I am grateful to **Mr. Sunil M. Nandam** , who gave an opportunity to work on projects at HighRadius Corporation and for their continuous support during the internship and for their patience, motivation and immense knowledge. They helped us and guided us throughout the internship and development.

I hereby take the privilege to express my gratitude to all the people who directly or indirectly involved in the execution of this work without whom this project would not have been a success.

I am also thankful to my seniors and team leads for their valuable guidance, support, and cooperation extended by them. Then I would like to thank my project team members for their kind cooperation, help and never-ending support.

I am also thankful to KIIT Bhubaneswar for providing me technical skills and facilities which proved to be very useful for our project.

Rakshita Sinha

1605213

Abstract

In Business to Business (B2B) scenarios large scale transactions are made on a daily basis for the products or services that a company supplies to its customers. For example, COMPANY ABC (CLIENT) is a company that sells Desktop and Personal Computers (PCs) for office work to various other companies such as Tata Motors, Amazon, HighRadius etc. Now, COMPANY ABC (CLIENT) is a huge company that sells thousands of its PCs to various other companies on a daily basis thus initiating large transactions where they are supposed to pay COMPANY ABC (CLIENT) for all the purchased goods. So, these various companies are customers of COMPANY ABC (CLIENT). COMPANY ABC (CLIENT) uses an Enterprise Resource Planning (ERP) software E.g. NetSuite or SAP to manage the invoices outstanding in the name of all its customers.

Once the customer (E.g. Tata Motors) makes the payment into the bank account of COMPANY ABC (CLIENT) two steps follow. First, the bank sends the payment file to the company i.e. COMPANY ABC (CLIENT) and second the customer (E.g. Tata Motors) sends the invoice details in the form of remittance to the company to list the invoices it has paid for in a particular payment. Then the payment and remittance information are matched by a Cash App Specialist and applied to the ERP system of COMPANY ABC (CLIENT) to clear the invoices open in the name of the corresponding customers.

When a company has thousands of large amount transactions made in a day by various customers, then posting cash manually, by employing a Cash Application specialist, makes the job extremely slow, time consuming, inefficient, expensive and yet prone to errors as human resources are involved.

This problem can be solved by using an automated Cash Application process wherein a software is customized for a particular client e.g. COMPANY ABC (CLIENT) can receive all the payments and remittances and apply cash based on the rules configured as per the requirements of the client. An automated cash application process is able to match payment and remittance at a much faster speed. As the cash application process has grown more and more complex, many companies have moved to an automated process, reducing the staff work load to reduce costs and work burnout. As a Cash App Consultant, the primary responsibility is to understand the AS IS process for particular client, as in their ERP and the rules they use to match payments and remittances to post cash and customize our own product to be able to apply cash in their system.

Moreover, if the customer submits checks and remittance stubs directly to the client, client has to scan these checks, clear their postings and send the payment information to the bank for actual payment. The RDC module makes this process easier by letting the client do exception handling in case of any capture issues. Also, it generates an image batch for CAA processing and an ICL file for payment processing in Bank.

OBJECTIVE

- The primary objective of this Project is to analyze, design, implement, test and manage the software product **Cash Application Automation (CAA)** and understand the role of **Remote Deposit Capture (RDC)** in CAA.
- The role includes the development and customization of the CAA software through an iterative model of Software Development Lifecycle (SDLC). The features of the master framework of CAA are changed or upgraded using a repository of rules via a cycle of Analysis, Design, Implementation and Testing phases.

A general framework of an iterative model of SDLC can be described by the Figure1.

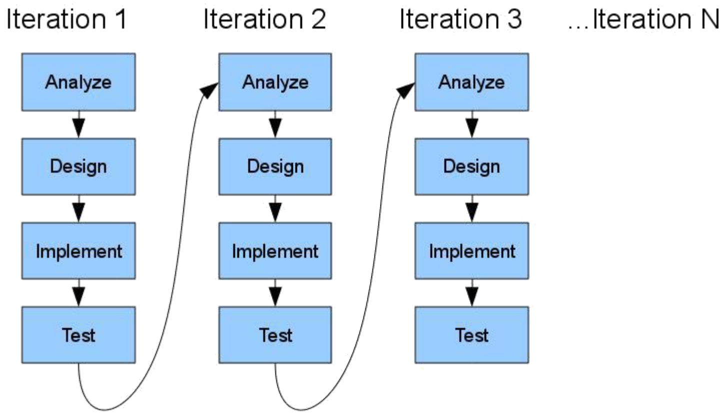


Figure 1 : Iterative model of SDLC

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER : 1

INTRODUCTION

In Business to Business (B2B) scenarios large scale transactions are made on a daily basis for the products or services that a company supplies to its customers. The company bills every customer as soon as the purchase is made. This is initiated by generating individual invoices against the specific customers for every purchase made by them- stating the expected payment amount and the due date by which it must be paid, along with other details.

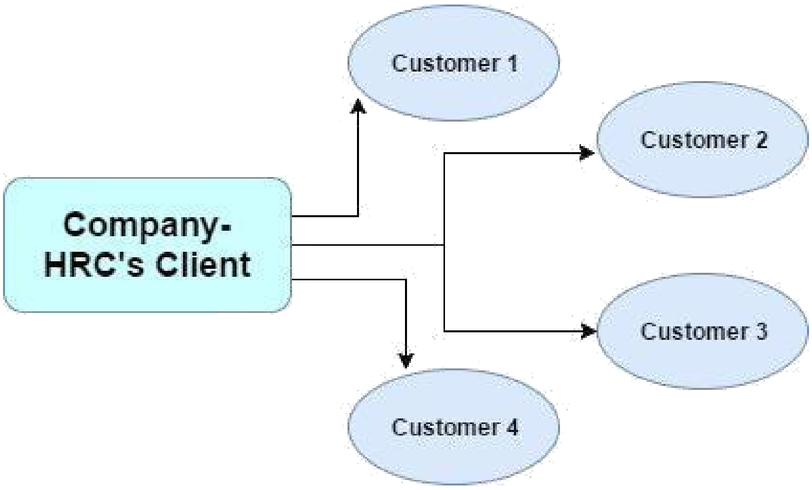


Fig. 2: Interaction of Client with HRC

The customers make the payments against the open invoices via Cheques or Electronic Fund Transfers (EFTs) into the bank account of the company. The bank then sends the payment information for all the payments made by each customer as payment files. A single payment made by the customer can be for the sum total of any number of invoices open against it. Hence, the customer also needs to send remittance information to the company giving the details of all the invoices it has paid for in a particular payment and also details of deductions (if any). The payments and remittances come from different sources- the bank and the customers, respectively and have to be matched with each other in order to close the open invoices outstanding in the customers' General Ledger (GL). Since these transactions are B2B they not only involve huge payments made across the globe but also the number of these transactions is very large for any manual system to keep account of.

In the past, the job of matching payments with corresponding remittances and then clearing the corresponding open invoices was done by full time Cash Applicants and involved valuable man hours. This process has been fully automated by a software product called Cash Application Automation (CAA) wherein the company becomes our client by buying the CAA product. Now the payment files from bank and the remittances from customers all flow into our CAA system, where all the matching and clearing of invoices is done. The output file thus generated is sent back to our client with the details of all the open invoices that can be closed by the company (Client).

Challenges faced in Manual Cash Application today and the solution proposed by automating the process with the use of Automation software is discussed in the introduction. This is followed by the description of the implementation phases and a survey of the past work done in this field. The terminology and the technologies used are discussed in detail thereafter. The important contribution of a Consultant is in the application of rules that justify and implement the business process. The result analysis section describes the comparative data analysis and the cost effectiveness of the end to end implementation of Cash Application Automation software.

1.1 Challenges in Manual Cash Application

The cash application process is highly manual, time-consuming, costly, and error-prone. Companies receive many forms of payment (checks, EFTs) across multiple banks. Customers provide inconsistent remittance detail in a variety of formats (paper, EDI, email attachments, online portals). In certain cases, remittance line items refer to alternative reference numbers that are not on the original invoice. Most company cash application programs require a clean, standardized remittance file to auto-apply cash. On top of this inefficiency, the cost of paying the bank to key in data from paper remittances can be very high and only partially address the problem. Audit and compliance issues arise when a company is sitting on unapplied or improperly applied cash, making accuracy and efficiency all the more critical.

When the payment is through checks, the credibility and correctness is also an area of concern.

1.2 Solution offered by RDC and CAA

Work Flow

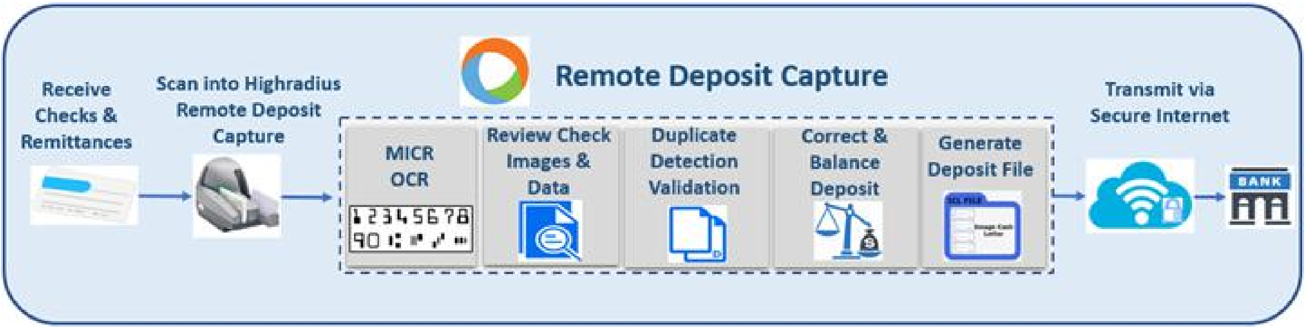

Remote Deposit Capture
Receive Checks & Remittances
Scan into Highradius Remote Deposit Capture
MICR OCR
12345678 90
Review Check Images & Data
Duplicate Detection Validation
Correct & Balance Deposit
Generate Deposit File
Transmit via Secure Internet
BANK

Fig. 3: RDC Work Flow

Remote Deposit Capture helps you to scan checks and remittances, and create digital deposits. These digital deposits can then be transmitted to your bank via a secure Internet. This eliminates the need to physically deliver the deposits. At the same time, these deposits will be applied to the corresponding invoices in the merchant's ERP system. In short, Remote Deposit Capture enables checks and remittances to be truncated and cleared digitally.

Benefits

- Reduced dependency on transportation cost
- Reduced processing cost
- Faster check clearance
- Browser-based Interface, which can process both deposits and remittances
- Integrated seamlessly into all Highradius Payments application
- Improved customer service

Operation Functions

With Highradius Remote Deposit Capture, an operator can perform the following functions:

- Scan checks and related remittances
- Correct the captured data from checks
- Review and balance the deposit amount
- Transmit deposits
- View previous deposit history

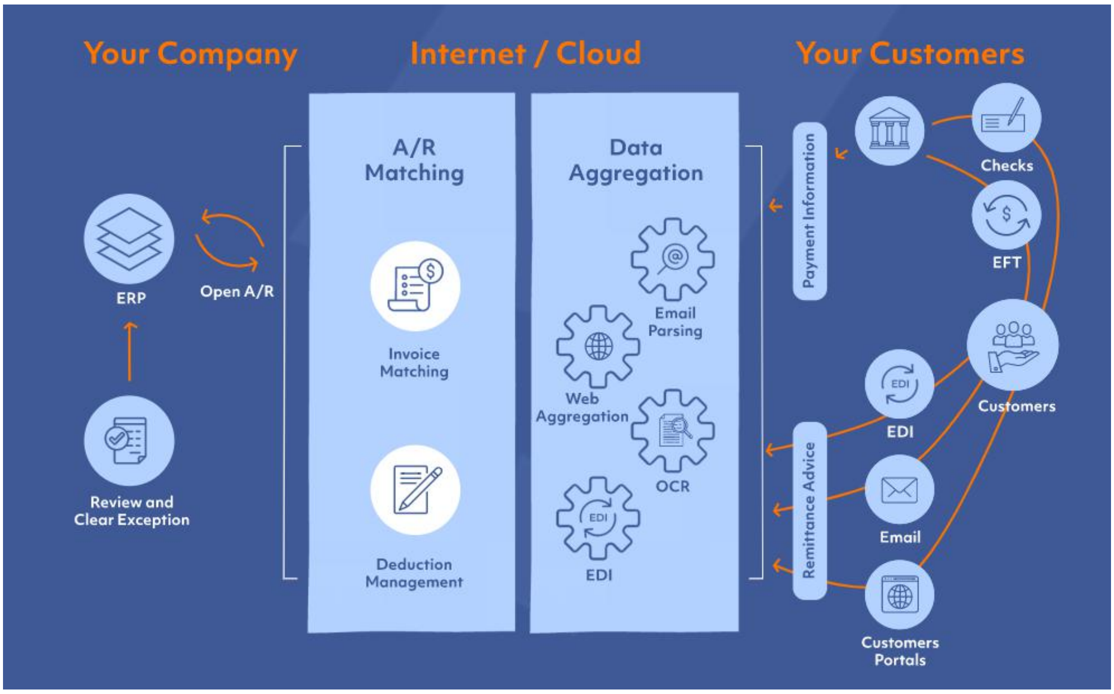


Fig. 4 : HighRadius CashApp Cloud

HighRadius Cash Application Cloud enables end-to-end automation of the cash application process. Other solutions address single parts of the process but do not provide a comprehensive approach. Our solution is holistic and does not require complex and costly on-premise installation or capital expenditure. A powerful rules engine matches patterns and cleanses data. A unique data aggregation engine parses email attachments and web portals, eliminating manual capture of remittance data and deduction backup. Easy integration with any ERP system maintains user interaction with standard ERP cash posting. The result is drastically reduced operating cost, near-perfect on-invoice hit rates, and better audit compliance.

Every client that is a user of this product undergoes an implementation cycle that involves a period of 3-4 months undertaken by a Cash App Consultant at HighRadius.

1.3 Basic Overview of the Implementation Phases

The cash App implementations are divided into 5 main tracks, viz. – Preparation, Blueprint, Realization, Testing, CutOver & Go-Live.

Ideally, the **Preparation** phase should start 2-4 weeks before planning to conduct the Blueprinting workshop. During this phase, Project is started by taking the Client's data and the data is analyzed.

Cash App is a very data dependent solution, so the idea is to build rules/agents to get as much automation as possible. It helps a lot to know client's data as much as possible before Blueprint starts especially remit formats, know little bit about client's customer.

As part of **Blueprinting**, first day or so – what is done is called AS-IS i.e. absorb client's business process– a day of cash app analyst is understood of a check payment and end to end walk through or take an electronic payment and end-to-end walk through. It might be little time consuming typically, it will last for a day or so. At end of AS-IS, workload analysis is done - one of our objectives is to increase productivity of client's team. So, as part of workload analysis, we quantify where the client is spending the most time and identify the highest opportunities to help it reduce the manual effort. After prioritising requirements, with the knowledge of AS-IS the TO-BE is proposed. This is presented to the client and, have few iterations back and forth and finally end result is the Process Design Document (PDD). So, PDD at that point is like bible for rest of the project. There might be few changes over the course but mostly will be as per the requirements.

After Blueprint is the **Realization** Phase. From the PDD, some key elements might require extra details. For instance, during PDD –capture has to take place from BAI2 file given by say Ban of America (BoA). But, during realization, it comes to notice couple of fields which the standard parser cannot extract. So, BAI2 parser has to be extended. This is put in a Technical Design Document and send it to client for a sign-off. Or the bank may be sending data in a proprietary format and a custom parser is required. The purpose of a Technical Design Document (TDD) is to ensure extra details are captured and agreed upon.

Then we get into **Testing**. Typically, there are 3 cycles of testing. These are System Integration Testing -1 where simulation of 1 days' worth of data is done. Ultimately, the objective is to simulate what happens in production. It won't be exact but try to set-up a simulation scenario which is close enough. So, just to clarify, this simulation is different from simulation analysis. Simulation is done and issues are fixed if any. Then, through cycle 2, a different data set or again the same data set. That's a decision to be taken during realization phase. So, by the time of cycle 2, most of the fixes are done as we have run through 5 days' worth of data and identified the issues. So, cycle 2 should be much smoother. Then by 4th week, once all fixes are done, we get into User Acceptance Testing (UAT). And then, at end of UAT, client will have to do a go/no-go decision– you are the client, you will make a decision given what you have seen in system is in-line with what it wanted. User training is also taken up during testing phase.

The last phase of the project is Cutover and go-live, which will happen over a 1 week period. Right after UAT, then preparation for production is started, and cutover activities and go-live is initiated.

The five phases of the implementation framework form one complete cycle of the iterative model of SDLC. Starting from requirement gathering and analysis to go live in the production phase, the end to end implementation of the CAA product is the sole scope of this project.

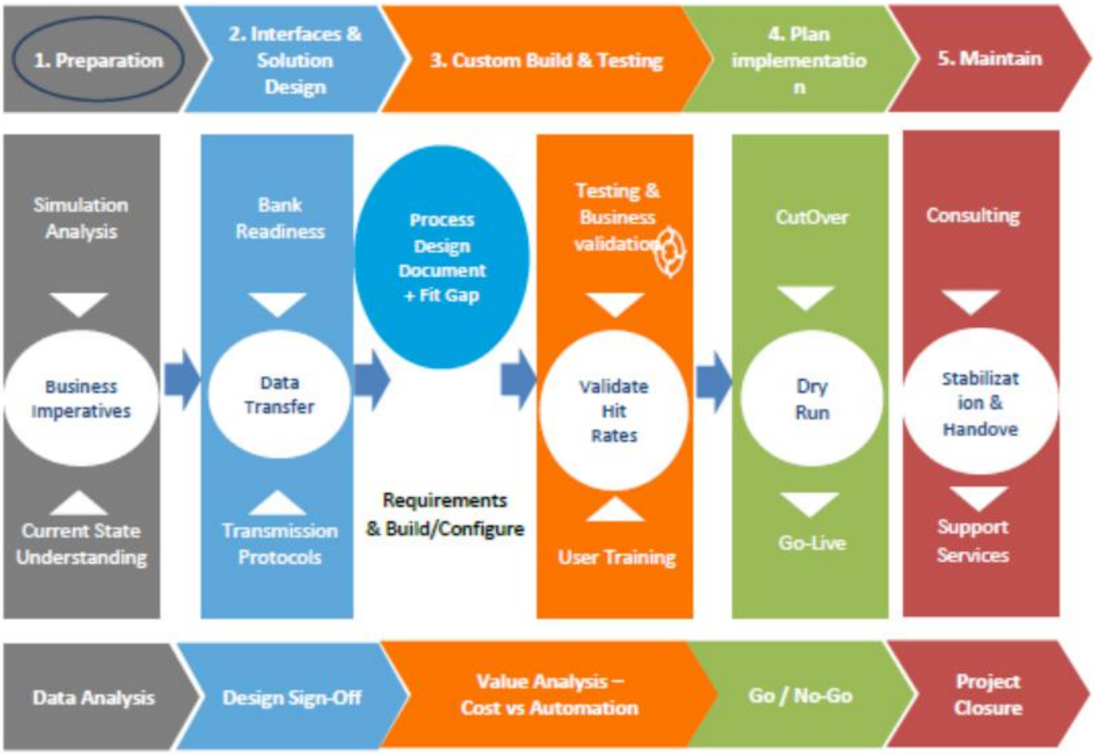


Figure 5 : Different Phases of the Project with Downstream of each phase at the bottom

CHAPTER : 2

LITERATURE SURVEY

After the post international financial crisis, the cash flow management has become the primary focus for Multinational enterprises. With the technological advancements in today’s era, it is necessary for the modern enterprises to re-examine the cash flow management from the internal perspective. Attempts have been made under the guidance of the application of internal control to build the scientific system, risk control system and value creation system of cash flow management along with the financial activities application guidance to create its new ideas, new concepts and new models [1]. But analysing the growing complexity of the Financial Supply Chain Management (FSCM) [2], we not only need systems and procedures to handle this complexity but also automated processes such as automation software to enable the cash flow management electronically with the least possible manual intervention.

Since the cash flow information can be applied well in the evaluation of an enterprise's financial situation, Li *et al.* in their paper [3] have regarded the cash profit, which is the information connotation of cash flow, as a basis of financial evaluation index system. Two early warning models based on cash profit and traditional accounting profit have been analysed here to give an early warning before a financial distress arises. Factoring Account Receivables is one technique that has been under research [4] to mitigate fluctuations in Cash flow for construction projects. In addition, the application of factoring has the advantage of facilitating financial management, instantly improving cash flow, enhancement of investment efficiency, avoiding extra loan procedures, improving credit rating, and transfer of financial risk

Financial disasters in private firms have led to increased emphasis on various forms of risk management, to include market risk management, operational risk management, and credit risk management. Financial institutions are motivated by the need to meet increased regulatory requirements for risk measurement and capital reserves. Wu *et al.* in their paper [5] describe and demonstrate a model to support risk management of accounts receivable. They have presented a credit scoring model to assess account credit worthiness. However, a continuous monitoring of the Credit extended by a Business to its Customer Businesses in large scale Business to Business (B2B) operations can prevent or mitigate any such financial crisis. This raises a demand for more automation software that can not only manage the cash flow in B2B operations but also handle large volume of transactions with ease on a day to day basis keeping a check on the Credit Management of the company.

Cash Application Automation (CAA), a software product of HighRadius, is one such attempt in this regard which duly addresses the problems of the Accounts Receivables section of the FSCM by automating the entire process of Cash Application for B2B scenario. The CAA software follows an iterative model of the Software Development Lifecycle (SDLC) wherein the entire lifecycle of development of the software is carried out based on the initial master framework developed in the first cycle of SDLC of CAA.

CHAPTER : 3

TERMINOLOGY AND TECHNOLOGIES INVOLVED

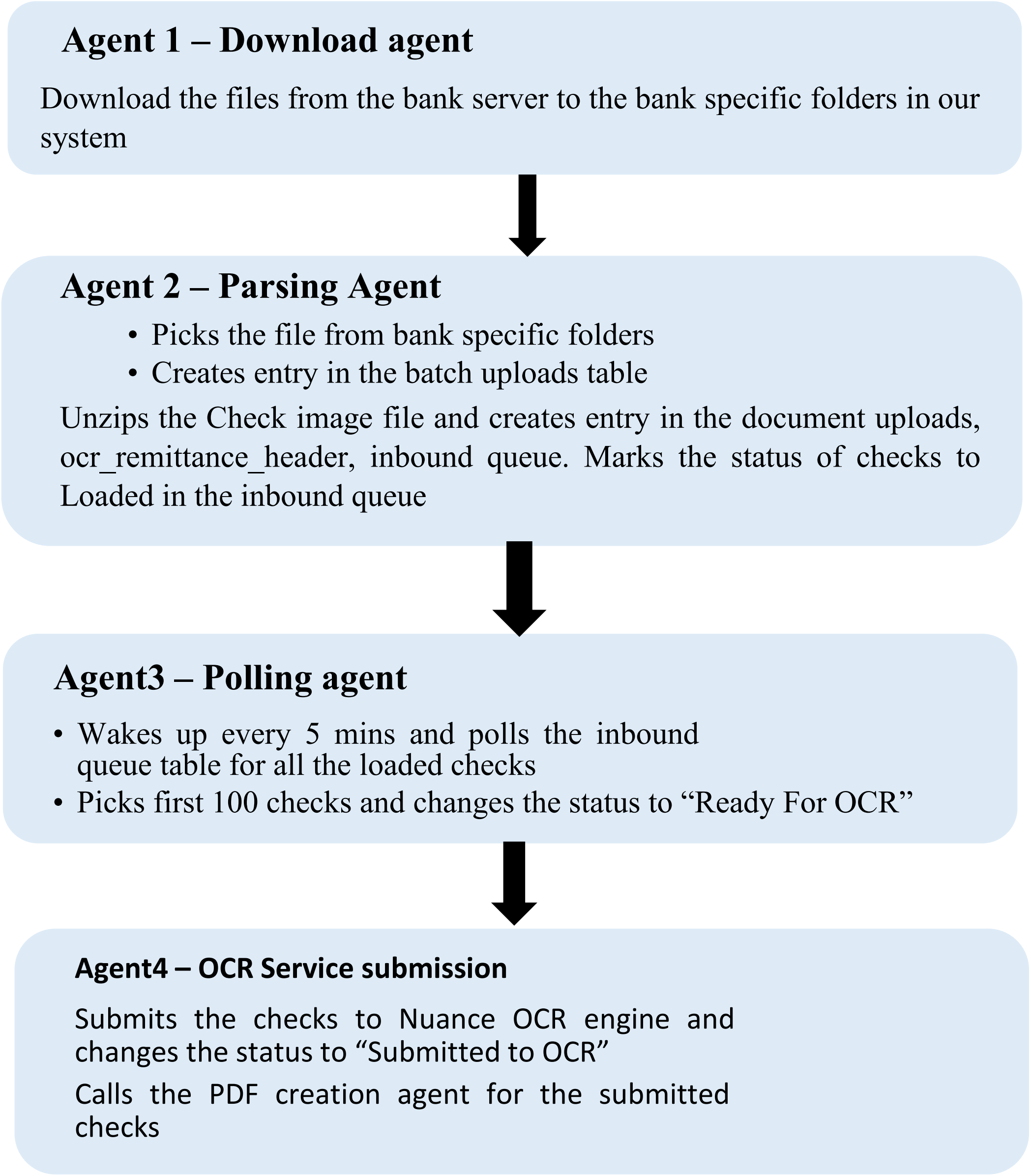
3.1 Terminology

The commonly used terms in CAA are:

- **Cash application:** Cash application is a part of the accounts receivable process that applies incoming payments to the correct customer accounts and receivable invoices.
- **Client**: Client is the one who is using our software.
- **Customer**: Client's end-customers. The companies that actually make Payments to our Clients.
- **Credit**: Credit is the trust maintained between a customer and client on the basis that a customer will make the payment for the purchase of goods or services at a future date.
- **Invoice**: A receipt of acknowledgement for the goods or services provided and their respective cost. Customer receives an invoice from the client specifying the goods dispatched or soled and the cost associated with the purchase.
- **Deduction**: When a customer pays less than the amount specified in the invoice because of any related issue with the goods received.
- **Accounts Receivables:** Accounts Receivables deals with the expected income from the sale of goods or services. (Client's perspective)
- **Customer Master:** Customer Master contains the details of the customers that do business with the client.
- **MICR:** Magnetic Ink Character Recognition. MICR number is a combination of
- Account number and IFSC code of the customer's account using which the customer pays the invoice.
- **Payment:** Money paid by Customer to Client for some product or service they bought and got billed for. Payment can be in the form of: Checks, ACH or Credit or Debit Card transactions.
- **Remittance:** A remittance advice is an explanation of the invoice & deduction details of the total Payment that a Customer made to a Client. Remittance can be in the form of: Paper backup to a Check that is scanned and made available as a TIF or PDF by Client's bank. It can also be in the form of EDI 820. In the form of an email with an attachment or body of the email. A new emerging trend is that the Customer provides an access to a website for Client to login and retrieve.

3.2 OCR Technology

OCR (Optical Character Recognition) technology is used to extract the remittance lines from the remittance information that is made available by the customer. The process flow of extracting remittance line items using OCR engine is described below:



Agent 1 – Download agent

Download the files from the bank server to the bank specific folders in our system

Agent 2 – Parsing Agent

- Picks the file from bank specific folders
- Creates entry in the batch uploads table

Unzips the Check image file and creates entry in the document uploads, ocr_remittance_header, inbound queue. Marks the status of checks to Loaded in the inbound queue

Agent3 – Polling agent

- Wakes up every 5 mins and polls the inbound queue table for all the loaded checks
- Picks first 100 checks and changes the status to "Ready For OCR"

Agent4 – OCR Service submission

Submits the checks to Nuance OCR engine and changes the status to "Submitted to OCR"

Calls the PDF creation agent for the submitted checks

Figure 6 : Process flow for Remittance Data Capture from emails

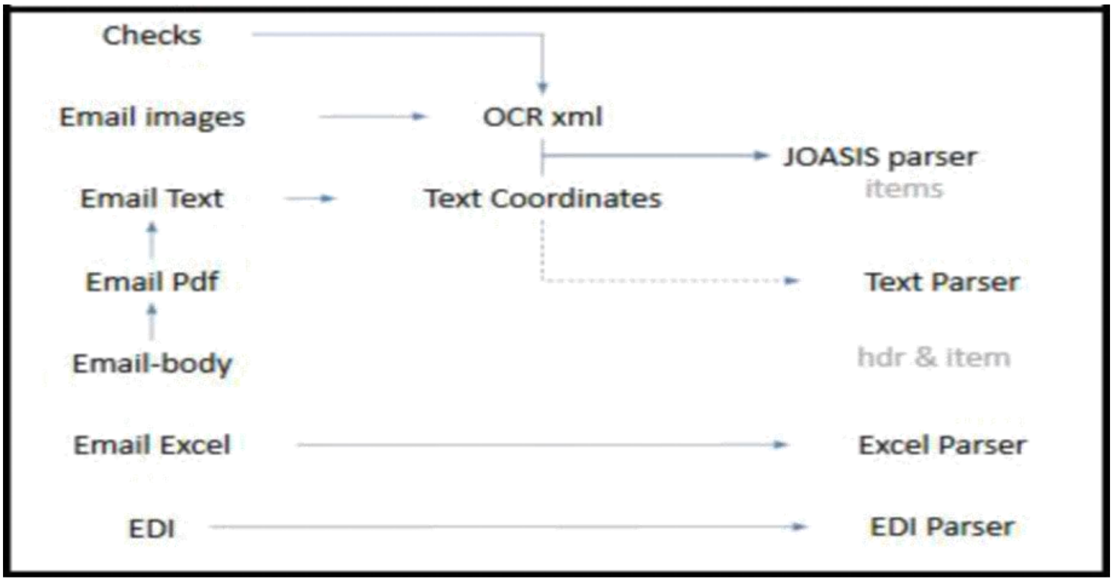
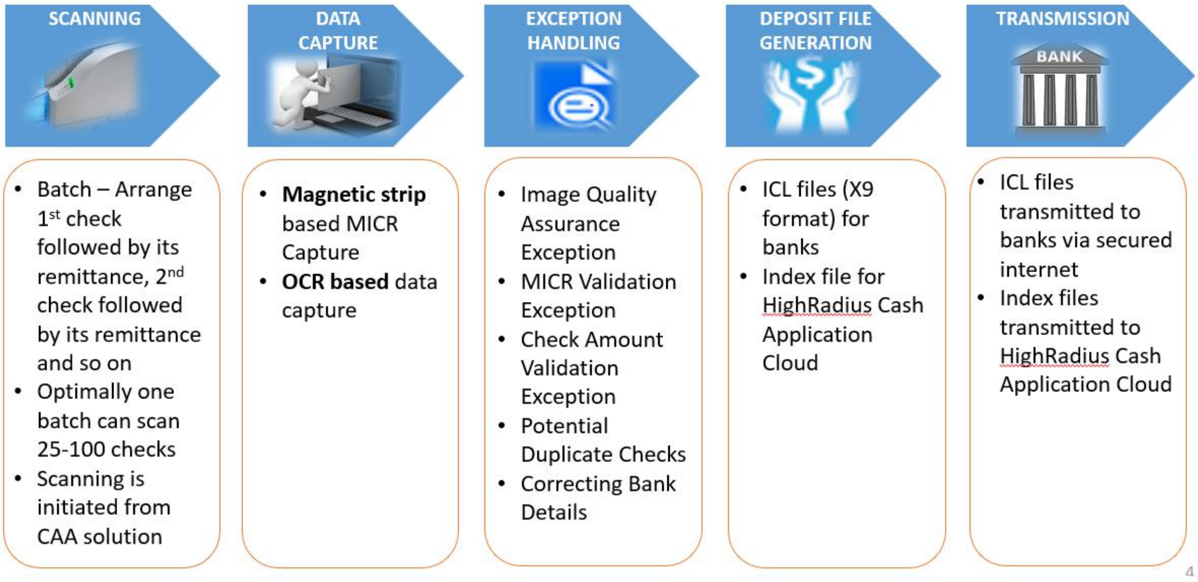


Figure 7 : Various parsers used for different remits



- Batch – Arrange 1st check followed by its remittance, 2nd check followed by its remittance and so on
- Optimally one batch can scan 25-100 checks
- Scanning is initiated from CAA solution

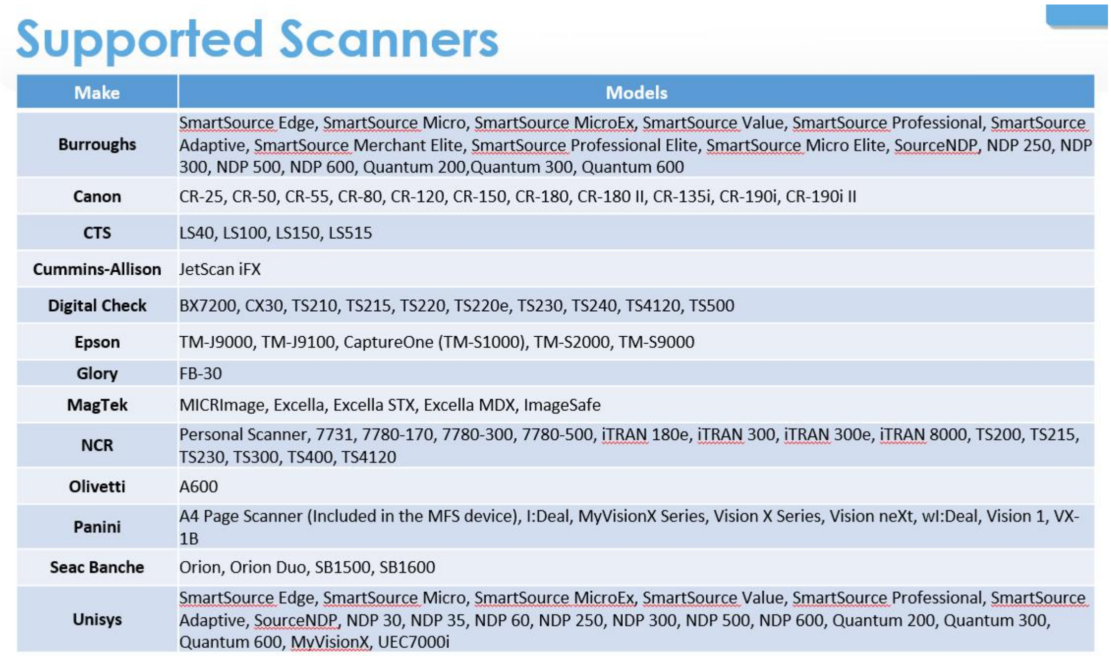
- **Magnetic strip** based MICR Capture
- **OCR based** data capture

- Image Quality Assurance Exception
- MICR Validation Exception
- Check Amount Validation Exception
- Potential Duplicate Checks
- Correcting Bank Details

- ICL files (X9 format) for banks
- Index file for HighRadius Cash Application Cloud

- ICL files transmitted to banks via secured internet
- Index files transmitted to HighRadius Cash Application Cloud

Figure 8 : Process flow for Payment and Remittance Data Capture by RDC



Supported Scanners

Make	Models
Burroughs	SmartSource Edge, SmartSource Micro, SmartSource MicroEx, SmartSource Value, SmartSource Professional, SmartSource Adaptive, SmartSource Merchant Elite, SmartSource Professional Elite, SmartSource Micro Elite, SourceNDP, NDP 250, NDP 300, NDP 500, NDP 600, Quantum 200,Quantum 300, Quantum 600
Canon	CR-25, CR-50, CR-55, CR-80, CR-120, CR-150, CR-180, CR-180 II, CR-135i, CR-190i, CR-190i II
CTS	LS40, LS100, LS150, LS515
Cummins-Allison	JetScan iFX
Digital Check	BX7200, CX30, TS210, TS215, TS220, TS220e, TS230, TS240, TS4120, TS500
Epson	TM-J9000, TM-J9100, CaptureOne (TM-S1000), TM-S2000, TM-S9000
Glory	FB-30
MagTek	MICRImage, Excella, Excella STX, Excella MDX, ImageSafe
NCR	Personal Scanner, 7731, 7780-170, 7780-300, 7780-500, iTRAN 180e, iTRAN 300, iTRAN 300e, iTRAN 8000, TS200, TS215, TS230, TS300, TS400, TS4120
Olivetti	A600
Panini	A4 Page Scanner (Included in the MFS device), I:Deal, MyVisionX Series, Vision X Series, Vision neXt, wI:Deal, Vision 1, VX-1B
Seac Banche	Orion, Orion Duo, SB1500, SB1600
Unisys	SmartSource Edge, SmartSource Micro, SmartSource MicroEx, SmartSource Value, SmartSource Professional, SmartSource Adaptive, SourceNDP, NDP 30, NDP 35, NDP 60, NDP 250, NDP 300, NDP 500, NDP 600, Quantum 200, Quantum 300, Quantum 600, MyVisionX, UEC7000i

Figure 9 : Various scanners supported by HRC for RDC

3.3 Relational database management system

A Relational Database Management System (RDBMS) is a Database Management System (DBMS) that is based on the relational model as invented by E. F. Codd, of IBM's San Jose Research Laboratory. In 2017, many of the databases in widespread use are based on the relational database model.

RDBMSs have been a common choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and other applications since the 1980s. Relational databases have often replaced legacy hierarchical databases and network databases because they are easier to understand and use. However, relational databases have received unsuccessful challenge attempts by object database management systems in the 1980s and 1990s (which were introduced trying to address the so-called object- relational impedance mismatch between relational databases and object-oriented application programs) and also by XML database management systems in the 1990s. Despite such attempts, RDBMSs keep most of the market share, which has also grown over the years.

According to DB-Engines, in 2016, the most widely used systems are Oracle, MySQL (open source), Microsoft SQL Server, PostgreSQL (open source), IBM DB2, Microsoft Access, and SQLite (open source).

According to research company Gartner, in 2011, the five leading commercial relational database vendors by revenue were Oracle (48.8%), IBM (20.2%), Microsoft (17.0%), SAP including Sybase (4.6%), and Teradata (3.7%).

According to Gartner, in 2008, the percentage of database sites using any given technology were (a given site may deploy multiple technologies):

- Oracle Database – 70%
- Microsoft SQL Server – 68%
- MySQL (Oracle Corporation) – 50%
- IBM DB2 – 39%
- IBM Informix – 18%
- SAP Sybase Adaptive Server Enterprise – 15%
- SAP Sybase IQ – 14%
- Teradata – 11%

The term "relational database" was invented by E. F. Codd at IBM in 1970. Codd introduced the term in his seminal paper "A Relational Model of Data for Large Shared Data Banks". In this paper and later papers, he defined what he meant by "relational". One well-known definition of what constitutes a relational database system is composed of Codd's 12 rules. However, many of the early implementations of the relational model did not conform to all of Codd's rules, so the term gradually came to describe a broader class of database systems, which at a minimum:

Present the data to the user as relations (a presentation in tabular form, i.e. as a *collection* of tables with each table consisting of a set of rows and columns);

Provide relational operators to manipulate the data in tabular form.

The first systems that were relatively faithful implementations of the relational model were from the University of Michigan; Micro DBMS (1969), the Massachusetts Institute of Technology; (1971), and from IBM UK Scientific Centre at Peterlee; IS1 (1970–72) and its follow on PRTV (1973–79). The first system sold as an RDBMS was Multics Relational Data Store, first sold in 1978. Others have been Ingres and IBM BS12.

The most common definition of an RDBMS is a product that presents a view of data as a collection of rows and columns, even if it is not based strictly upon relational theory. By this definition, RDBMS products typically implement some but not all of Codd's 12 rules.

A second school of thought argues that if a database does not implement all of Codd's rules (or the current understanding on the relational model, as expressed by Christopher J Date, Hugh Darwen and others), it is not relational. This view, shared by many theorists and other strict adherents to Codd's principles, would disqualify most DBMSs as not relational. For clarification, they often refer to some RDBMSs as truly-relational database management systems (TRDBMS), naming others pseudo-relational database management systems (PRDBMS).

As of 2009, most commercial relational DBMS employ SQL as their query language.

Alternative query languages have been proposed and implemented, notably the pre-1996 implementation of Ingres QUEL.

3.4 SQL

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is a language to operate databases; it includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an **ANSI** (American National Standards Institute) standard language, but there are many different versions of the SQL language.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

Also, they are using different dialects, such as −

- MS SQL Server using T-SQL,
- Oracle using PL/SQL,
- MS Access version of SQL is called JET SQL (native format) etc.

Why SQL?

SQL is widely popular because it offers the following advantages −

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

A Brief History of SQL

1970 − Dr. Edgar F. "Ted" Codd of IBM is known as the father of relational databases. He described a relational model for databases.

1974 − Structured Query Language appeared.

1978 − IBM worked to develop Codd's ideas and released a product named
System/R.

1986 − IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Software which later came to be known as Oracle.

SQL Process

When you are executing an SQL command for any RDBMS, the system determines the best way to carry out your request and SQL engine figures out how to interpret the task.

There are various components included in this process are-

- Query Dispatcher
- Optimization Engines
- Classic Query Engine
- SQL Query Engine, etc.

A classic query engine handles all the non-SQL queries, but a SQL query engine won't handle logical files.

Following is a simple diagram showing the SQL Architecture −

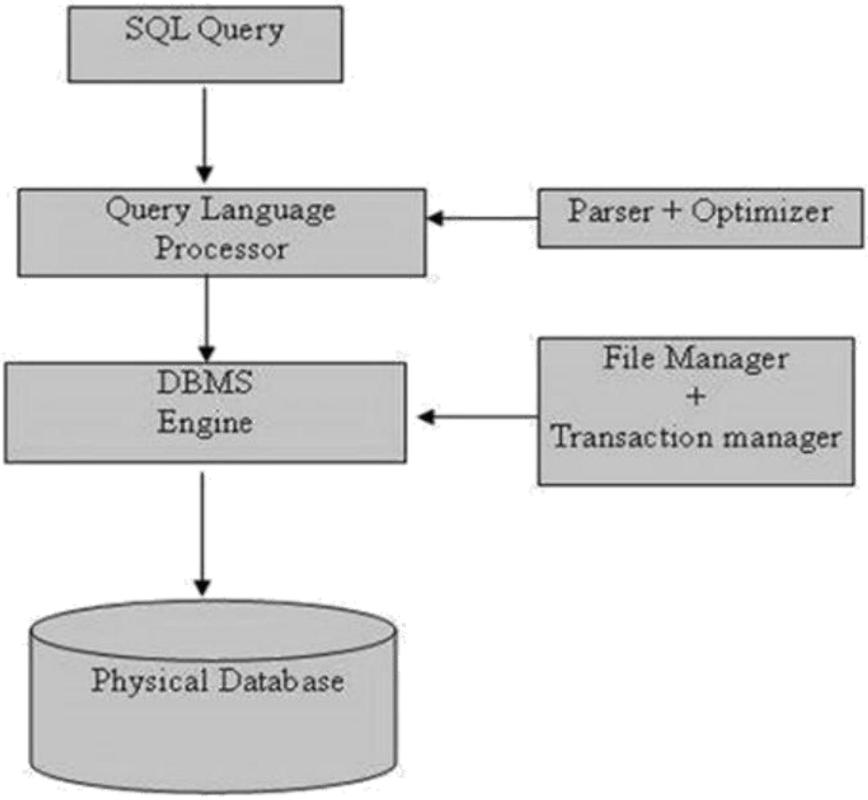
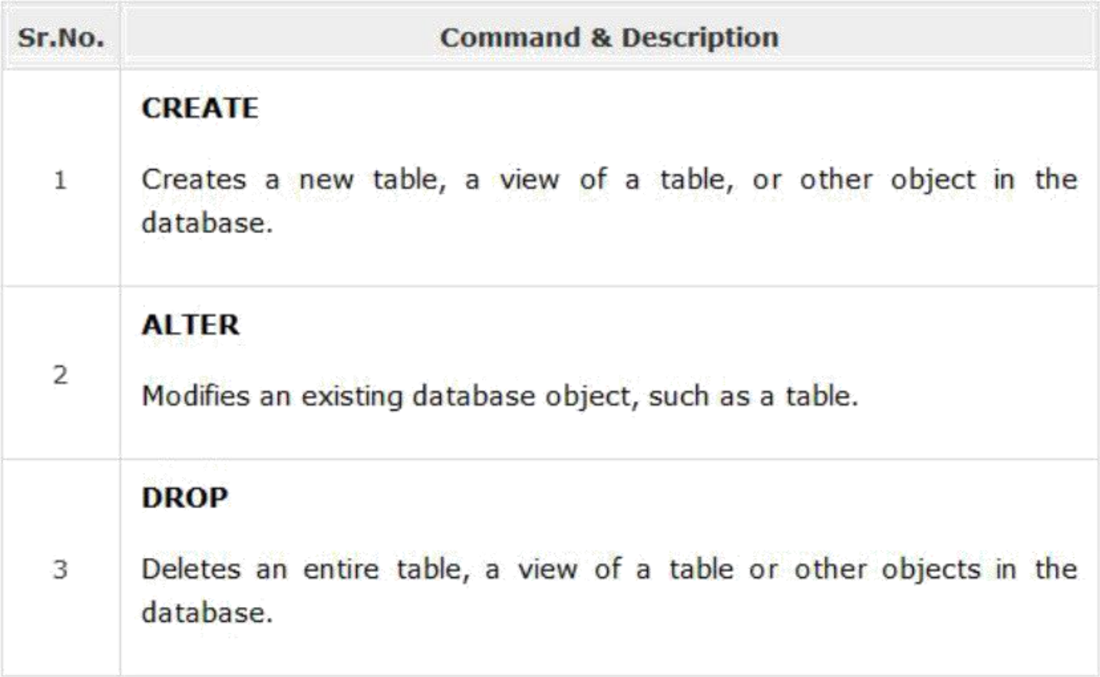


Figure 9 : Use of SQL in access of database

SQL Commands

The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into the following groups based on their nature −

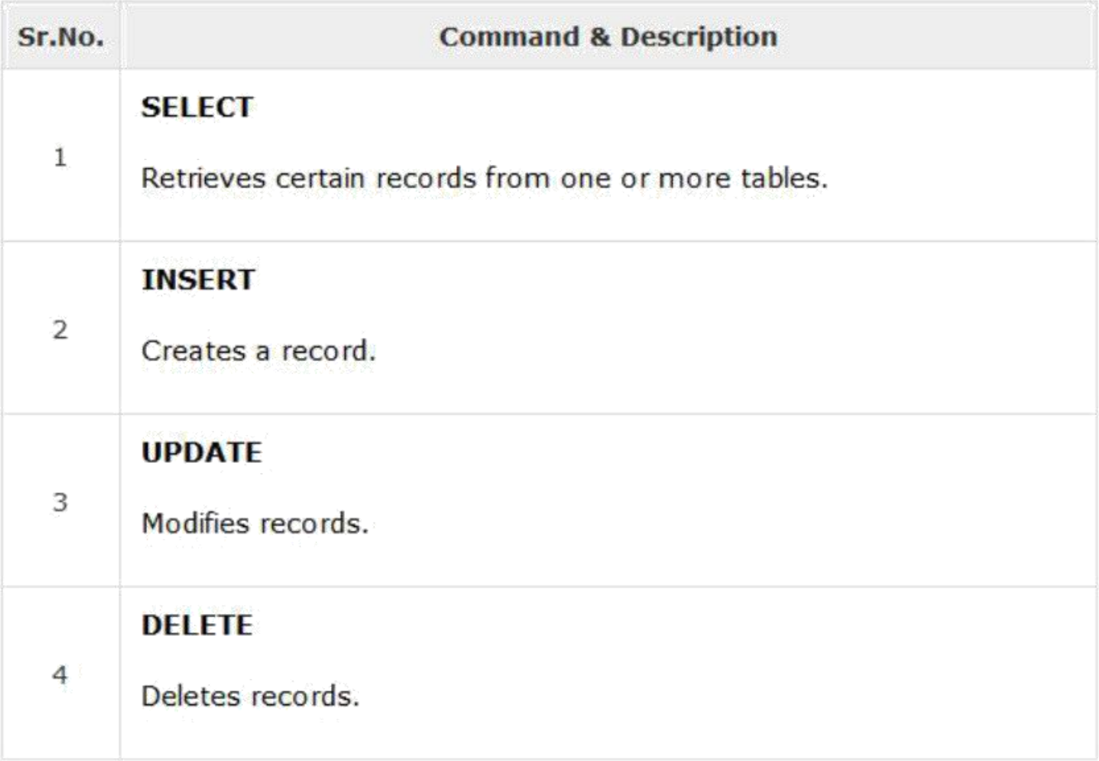
DDL - Data Definition Language



Sr.No.	Command & Description
1	**CREATE** Creates a new table, a view of a table, or other object in the database.
2	**ALTER** Modifies an existing database object, such as a table.
3	**DROP** Deletes an entire table, a view of a table or other objects in the database.

Table 1: Data Definition Language

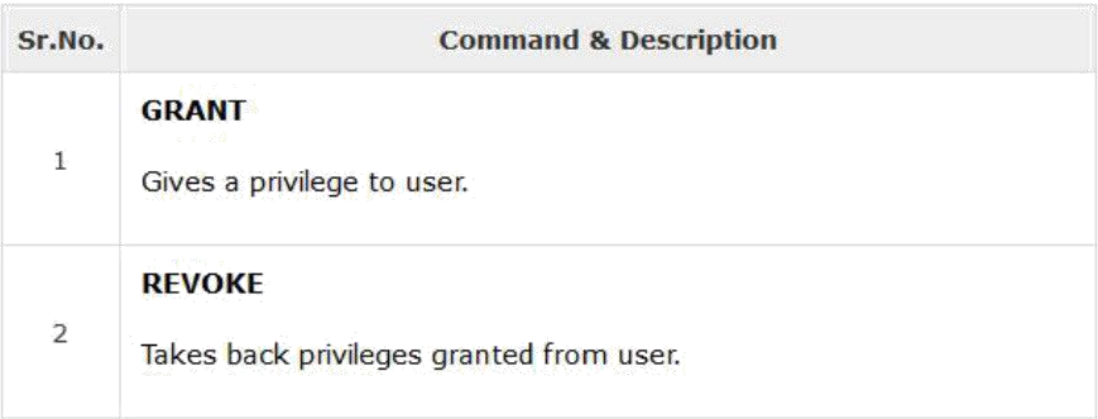
DML - Data Manipulation Language



Sr.No.	Command & Description
1	**SELECT** Retrieves certain records from one or more tables.
2	**INSERT** Creates a record.
3	**UPDATE** Modifies records.
4	**DELETE** Deletes records.

Table 2: Data Manipulation Language

DCL - Data Control Language



Sr.No.	Command & Description
1	**GRANT** Gives a privilege to user.
2	**REVOKE** Takes back privileges granted from user.

Table 3 : Data Control Language

CHAPTER : 4

REQUIREMENT ANALYSIS

Cash Application is the process of matching of payment amount against the remittance line items for a particular customer and clearing the open invoices in the company's or our client's ERP. In business terminology, Cash application is a part of the accounts receivable process that applies incoming payments to the correct customer accounts and receivable invoices

In the past, the job of matching payments with corresponding remittances and then clearing the corresponding open invoices was done by full time Cash Applicants and involved valuable man hours.

The business process involved in the manual application of cash or payments by a Client is termed as **AS-IS state** of the company. The Cash Application Specialist or the Cash Applicant manually clears the invoice in the Company's ERP on receiving and matching the remittance and payment files from the customers and banks respectively.

4.1 AS IS State of Cash Application

As discussed above, Cash Application is the process of applying payments to open receivables. This includes payments made against invoices, deductions made against invoices, non-invoice related deductions, credits claimed by customers and repayments against denied deductions.

We shall take a Use Case Scenario of a COMPANY ABC (CLIENT) to explain the Cash Application Process carried out by the CAA software using the Cash Application Cloud. The AS IS diagram shown in Fig. 8 shows the AS IS state of Cash Application process of COMPANY ABC where all the cash postings are done manually.

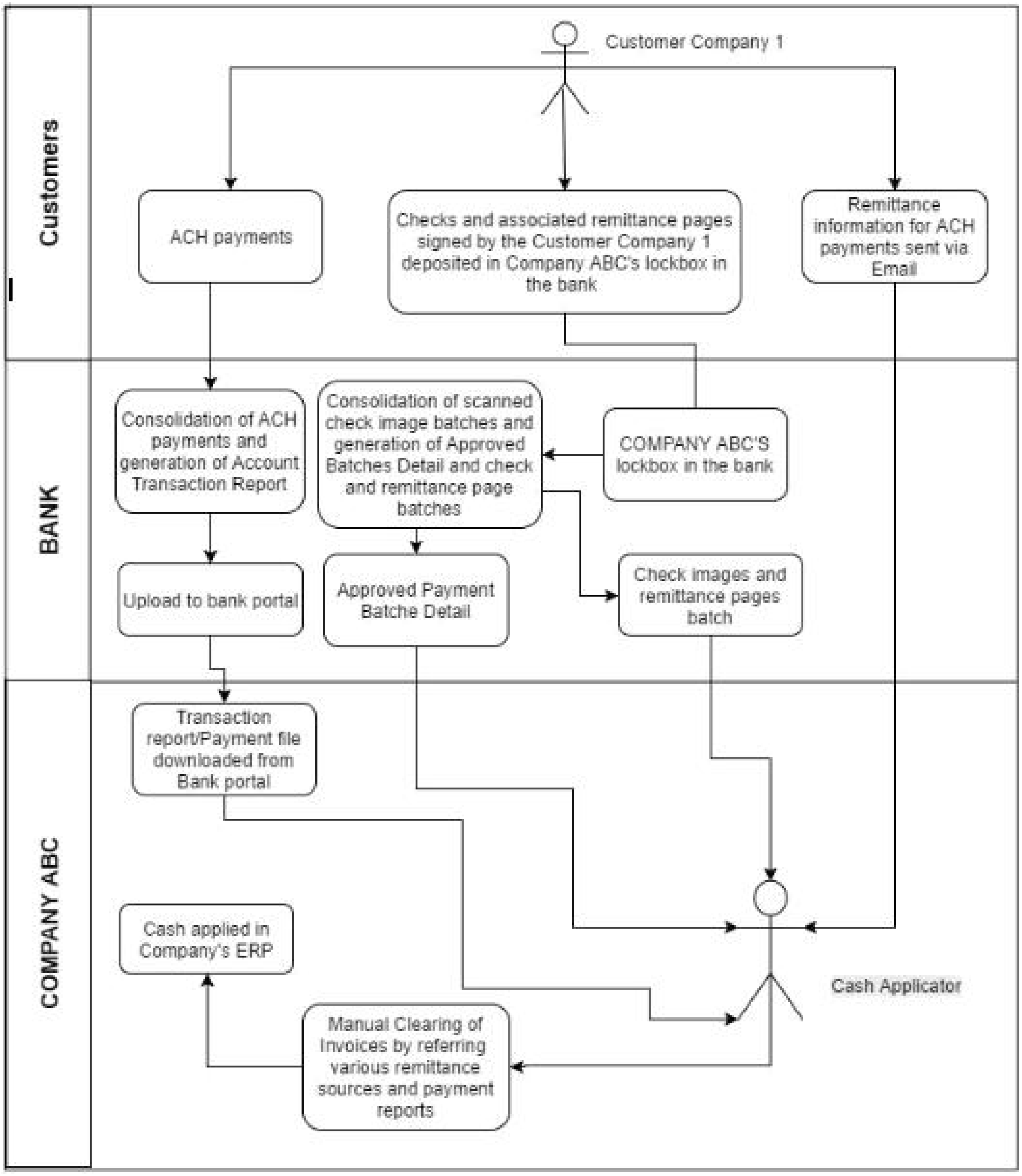

Customers
BANK
COMPANY ABC
Customer Company 1
ACH payments
Checks and associated remittance pages signed by the Customer Company 1 deposited in Company ABC's lockbox in the bank
Remittance information for ACH payments sent via Email
Consolidation of ACH payments and generation of Account Transaction Report
Consolidation of scanned check image batches and generation of Approved Batches Detail and check and remittance page batches
COMPANY ABC'S lockbox in the bank
Upload to bank portal
Approved Payment Batche Detail
Check images and remittance pages batch
Transaction report/Payment file downloaded from Bank portal
Cash applied in Company's ERP
Manual Clearing of Invoices by referring various remittance sources and payment reports
Cash Applicator

Figure 10 : AS IS state of Company ABC

Lockbox Payment processing:

US lockbox payments are deposited to multiple Bank Lockboxes. Banks key in the Check Remittance and send a BAI2 file with the Payment and Remittance details. These files are loaded into the ERP through either automatic or manual process depending upon the banks' setup with the Client.

1. **Manual:** Banks will send BAI2 file through email or Users will download BAI2 file from bank portals. Then these files will be uploaded in the ERP by users manually

2. **Automatic:** Host to Host connection has been setup with some banks through which BAI2 files will be posted in the ERP automatically.

EFT Payment Processing:

The EFT files processing could be automatic, Semi-Automatic or Manual depending upon the banks' setup with the Client:

1. **Automatic:** The partner banks send payment files e.g. MT940 files to bank hub or bank portal. Through B2B, these files will be posted to ERP automatically
2. **Semi-Automatic:** Users will download these payment files from bank portals and upload those files into ERP. Invoices which are provided will be posted automatically. Users will manually handle remaining payments.
3. **Manual:** Users will download bank statements from bank portals and then manually post in ERP.

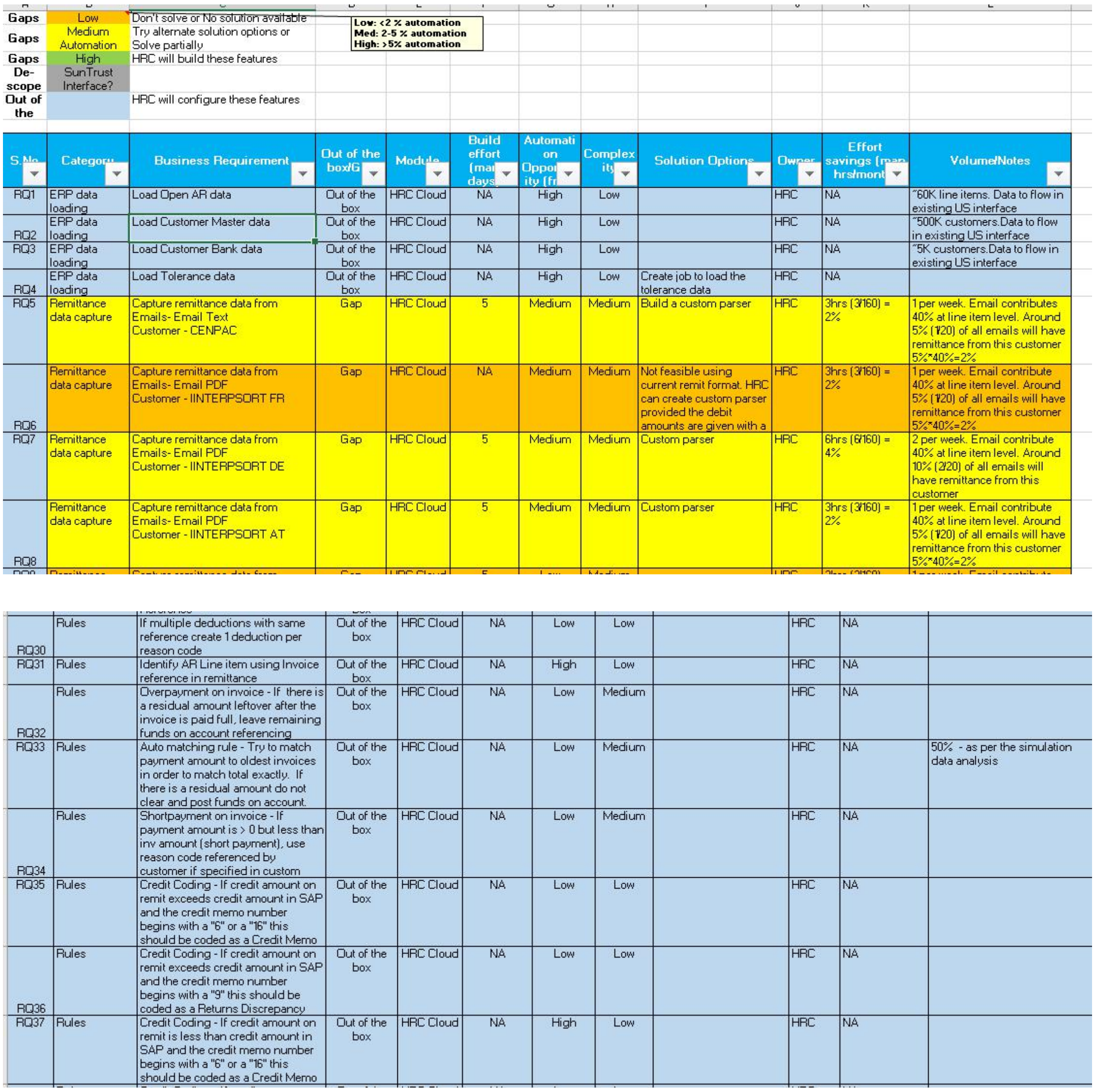
CHAPTER : 5

SYSTEM DESIGN

After having a complete idea of the requirement of the client, consultants try to design the product accordingly.

5.1 Blueprinting

Consultants try to come up with a document which has list of all requirement classified on the basis of what is already there and wht is to be developed.
This is called a fit gap document.



Gaps	Low	Don't solve or No solution available
Gaps	Medium Automation	Try alternate solution options or Solve partially
Gaps	High	HRC will build these features
De-scope	SunTrust Interface?	
Out of the		HRC will configure these features

Low: <2 % automation
Med: 2-5 % automation
High: >5% automation

S.No	Category	Business Requirement	Out of the box/G	Module	Build effort (ma days	Automation on Oppor ity (fr	Complexity	Solution Options	Owner	Effort savings (man hrs/mont	Volume/Notes
RQ1	ERP data loading	Load Open AR data	Out of the box	HRC Cloud	NA	High	Low		HRC	NA	~60K line items. Data to flow in existing US interface
RQ2	ERP data loading	Load Customer Master data	Out of the box	HRC Cloud	NA	High	Low		HRC	NA	~500K customers.Data to flow in existing US interface
RQ3	ERP data loading	Load Customer Bank data	Out of the box	HRC Cloud	NA	High	Low		HRC	NA	~5K customers.Data to flow in existing US interface
RQ4	ERP data loading	Load Tolerance data	Out of the box	HRC Cloud	NA	High	Low	Create job to load the tolerance data	HRC	NA	
RQ5	Remittance data capture	Capture remittance data from Emails- Email Text Customer - CENPAC	Gap	HRC Cloud	5	Medium	Medium	Build a custom parser	HRC	3hrs (3/160) = 2%	1 per week. Email contributes 40% at line item level. Around 5% (1/20) of all emails will have remittance from this customer 5%*40%=2%
RQ6	Remittance data capture	Capture remittance data from Emails- Email PDF Customer - IINTERPSORT FR	Gap	HRC Cloud	NA	Medium	Medium	Not feasible using current remit format. HRC can create custom parser provided the debit amounts are given with a	HRC	3hrs (3/160) = 2%	1 per week. Email contribute 40% at line item level. Around 5% (1/20) of all emails will have remittance from this customer 5%*40%=2%
RQ7	Remittance data capture	Capture remittance data from Emails- Email PDF Customer - IINTERPSORT DE	Gap	HRC Cloud	5	Medium	Medium	Custom parser	HRC	6hrs (6/160) = 4%	2 per week. Email contribute 40% at line item level. Around 10% (2/20) of all emails will have remittance from this customer
RQ8	Remittance data capture	Capture remittance data from Emails- Email PDF Customer - IINTERPSORT AT	Gap	HRC Cloud	5	Medium	Medium	Custom parser	HRC	3hrs (3/160) = 2%	1 per week. Email contribute 40% at line item level. Around 5% (1/20) of all emails will have remittance from this customer 5%*40%=2%
RQ30	Rules	If multiple deductions with same reference create 1 deduction per reason code	Out of the box	HRC Cloud	NA	Low	Low		HRC	NA	
RQ31	Rules	Identify AR Line item using Invoice reference in remittance	Out of the box	HRC Cloud	NA	High	Low		HRC	NA	
RQ32	Rules	Overpayment on invoice - If there is a residual amount leftover after the invoice is paid full, leave remaining funds on account referencing	Out of the box	HRC Cloud	NA	Low	Medium		HRC	NA	
RQ33	Rules	Auto matching rule - Try to match payment amount to oldest invoices in order to match total exactly. If there is a residual amount do not clear and post funds on account.	Out of the box	HRC Cloud	NA	Low	Medium		HRC	NA	50% - as per the simulation data analysis
RQ34	Rules	Shortpayment on invoice - If payment amount is > 0 but less than inv amount (short payment), use reason code referenced by customer if specified in custom	Out of the box	HRC Cloud	NA	Low	Medium		HRC	NA	
RQ35	Rules	Credit Coding - If credit amount on remit exceeds credit amount in SAP and the credit memo number begins with a "6" or a "16" this should be coded as a Credit Memo	Out of the box	HRC Cloud	NA	Low	Low		HRC	NA	
RQ36	Rules	Credit Coding - If credit amount on remit exceeds credit amount in SAP and the credit memo number begins with a "9" this should be coded as a Returns Discrepancy	Out of the box	HRC Cloud	NA	Low	Low		HRC	NA	
RQ37	Rules	Credit Coding - If credit amount on remit is less than credit amount in SAP and the credit memo number begins with a "6" or a "16" this should be coded as a Credit Memo	Out of the box	HRC Cloud	NA	High	Low		HRC	NA	

Figure 11 : Fit Gap Document of Company ABC

5.2 TO-BE State of the Project

Once the process has been fully automated by our software product called Cash Application Automation (CAA), the COMPANYABC becomes our client by buying CAA product.

Now the payment files from bank and the remittances from customers all flow into our CAA system, where all the matching and clearing of invoices is done.

The output file thus generated is sent back to our client with the details of all the open invoices that can be closed by the company (Client) in its ERP system. The Cash posting thus becomes fully automated, extremely fast and resource effective.

The TO BE state of Cash Application system in place for COMPANY ABC after using CAA is shown in Fig. 10.

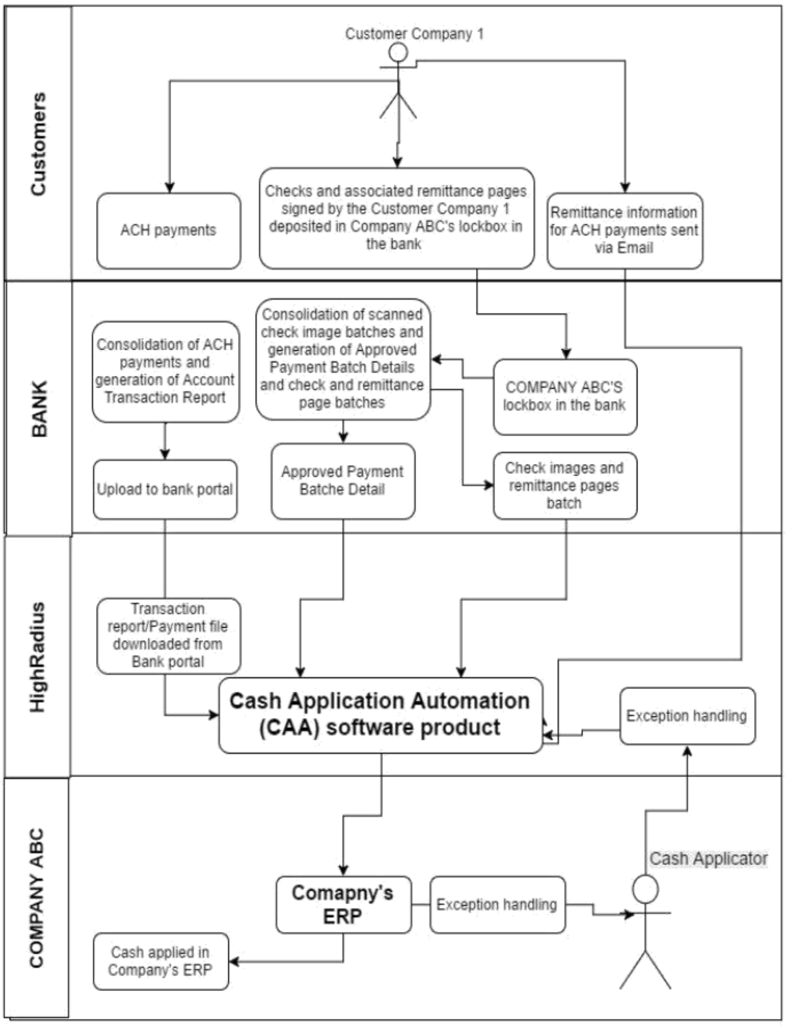
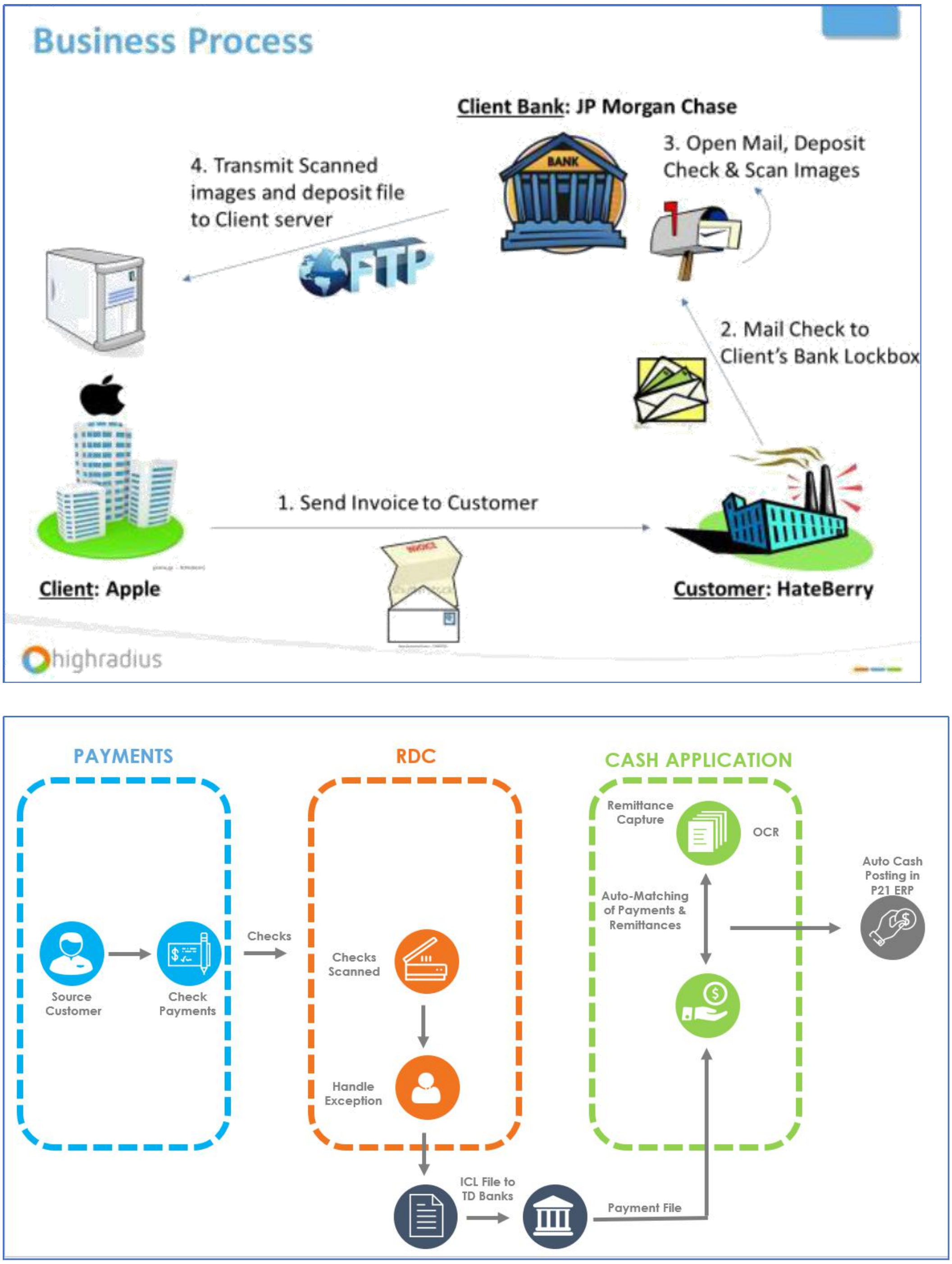

Customers
Customer Company 1
ACH payments
Checks and associated remittance pages signed by the Customer Company 1 deposited in Company ABC's lockbox in the bank
Remittance information for ACH payments sent via Email
BANK
Consolidation of ACH payments and generation of Account Transaction Report
Consolidation of scanned check image batches and generation of Approved Payment Batch Details and check and remittance page batches
COMPANY ABC'S lockbox in the bank
Upload to bank portal
Approved Payment Batche Detail
Check images and remittance pages batch
HighRadius
Transaction report/Payment file downloaded from Bank portal
Cash Application Automation (CAA) software product
Exception handling
COMPANY ABC
Cash Applicator
Comapny's ERP
Exception handling
Cash applied in Company's ERP

Figure 12 : TO BE state of Cash Application after CAA implementation

After the Cash App Automation system is in place for COMPANY ABC (CLIENT) the expected future state of the Cash Application is also presented by the illustration in Fig. 11:



Business Process

Client Bank: JP Morgan Chase

4. Transmit Scanned images and deposit file to Client server

3. Open Mail, Deposit Check & Scan Images

2. Mail Check to Client's Bank Lockbox

1. Send Invoice to Customer

Client: Apple

Customer: HateBerry

Figure 13 : Illustration of business process for check payments after CAA implementation

CHAPTER : 6

SYSTEM TESTING

6.1 Testing Approach

- Testing is based on the entire day’s payment files (Batch) using real production data
- Objective is to progressively improve posting with each batch testing
- Each batch is tested, results are analyzed, opportunities for improvement identified, and fixes are applied
- Issues are fixed with priority based on business value
- Same process is repeated in each batch

6.2 Testing Process Flow

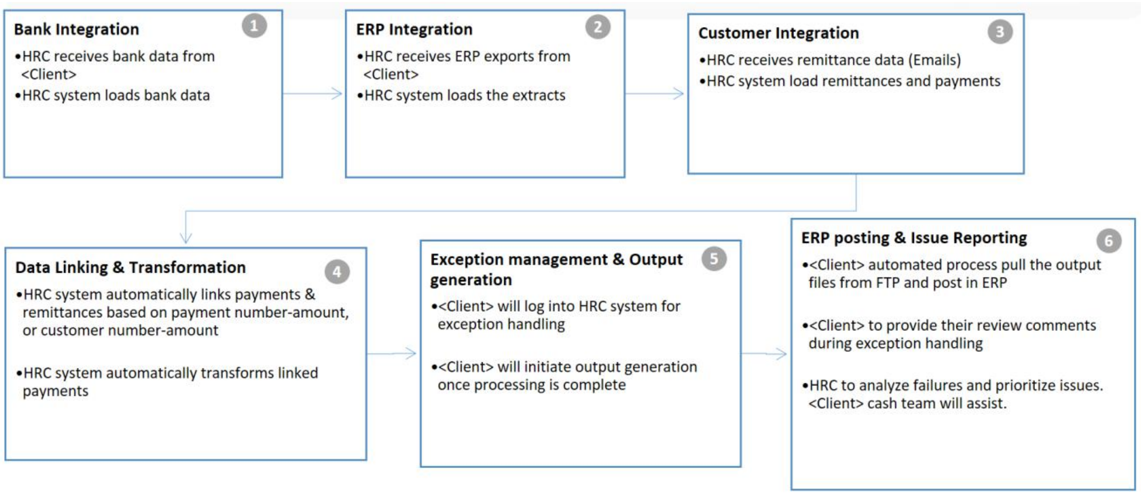
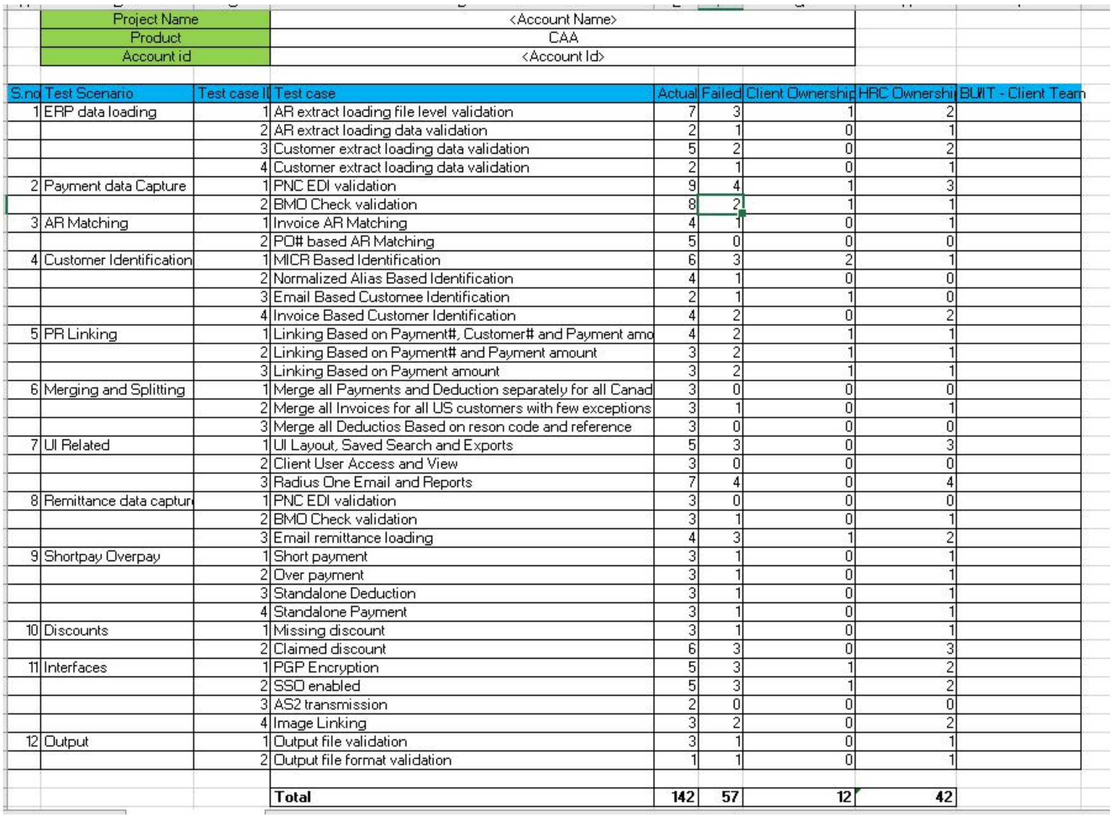

Bank Integration
1
•HRC receives bank data from <Client>
•HRC system loads bank data
ERP Integration
2
•HRC receives ERP exports from <Client>
•HRC system loads the extracts
Customer Integration
3
•HRC receives remittance data (Emails)
•HRC system load remittances and payments
Data Linking & Transformation
4
•HRC system automatically links payments & remittances based on payment number-amount, or customer number-amount
•HRC system automatically transforms linked payments
Exception management & Output generation
5
•<Client> will log into HRC system for exception handling
•<Client> will initiate output generation once processing is complete
ERP posting & Issue Reporting
6
•<Client> automated process pull the output files from FTP and post in ERP
•<Client> to provide their review comments during exception handling
•HRC to analyze failures and prioritize issues. <Client> cash team will assist.

Figure 14 : Testing Process Flow of CAA implementation

6.3 Testing Script

Testing is done based on the following test scenarios and test cases

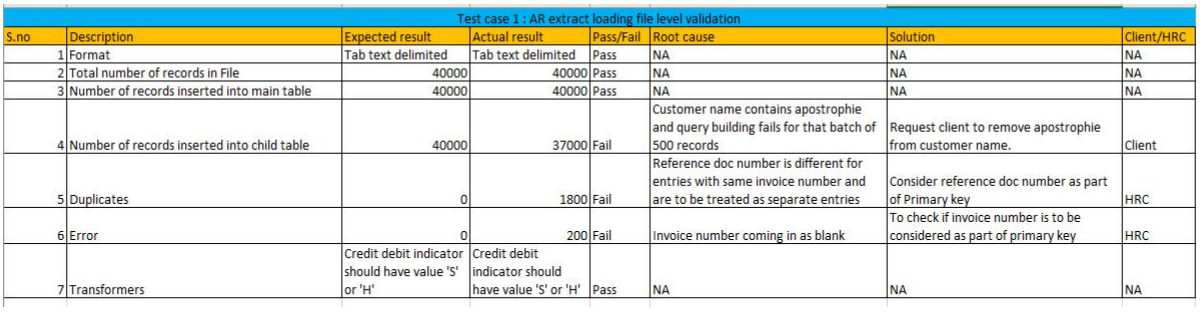


Project Name	<Account Name>
Product	CAA
Account id	<Account Id>

S.no	Test Scenario	Test case I	Test case	Actual	Failed	Client Ownership	HRC Ownershi	BU/IT - Client Team
1	ERP data loading	1	AR extract loading file level validation	7	3	1	2	
		2	AR extract loading data validation	2	1	0	1	
		3	Customer extract loading data validation	5	2	0	2	
		4	Customer extract loading data validation	2	1	0	1	
2	Payment data Capture	1	PNC EDI validation	9	4	1	3	
		2	BMO Check validation	8	2	1	1	
3	AR Matching	1	Invoice AR Matching	4	1	0	1	
		2	PO# based AR Matching	5	0	0	0	
4	Customer Identification	1	MICR Based Identification	6	3	2	1	
		2	Normalized Alias Based Identification	4	1	0	0	
		3	Email Based Customee Identification	2	1	1	0	
		4	Invoice Based Customer Identification	4	2	0	2	
5	PR Linking	1	Linking Based on Payment#, Customer# and Payment amo	4	2	1	1	
		2	Linking Based on Payment# and Payment amount	3	2	1	1	
		3	Linking Based on Payment amount	3	2	1	1	
6	Merging and Splitting	1	Merge all Payments and Deduction separately for all Canad	3	0	0	0	
		2	Merge all Invoices for all US customers with few exceptions	3	1	0	1	
		3	Merge all Deductios Based on reson code and reference	3	0	0	0	
7	UI Related	1	UI Layout, Saved Search and Exports	5	3	0	3	
		2	Client User Access and View	3	0	0	0	
		3	Radius One Email and Reports	7	4	0	4	
8	Remittance data captur	1	PNC EDI validation	3	0	0	0	
		2	BMO Check validation	3	1	0	1	
		3	Email remittance loading	4	3	1	2	
9	Shortpay Overpay	1	Short payment	3	1	0	1	
		2	Over payment	3	1	0	1	
		3	Standalone Deduction	3	1	0	1	
		4	Standalone Payment	3	1	0	1	
10	Discounts	1	Missing discount	3	1	0	1	
		2	Claimed discount	6	3	0	3	
11	Interfaces	1	PGP Encryption	5	3	1	2	
		2	SSO enabled	5	3	1	2	
		3	AS2 transmission	2	0	0	0	
		4	Image Linking	3	2	0	2	
12	Output	1	Output file validation	3	1	0	1	
		2	Output file format validation	1	1	0	1	
			Total	**142**	**57**	**12**	**42**	

Test cases for ERP Data Loading

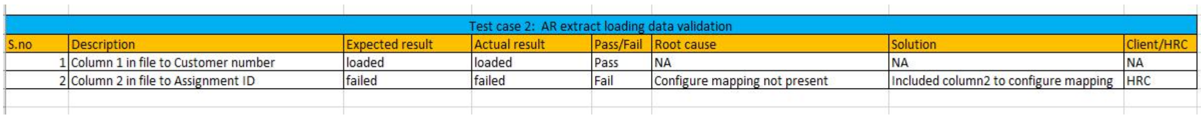
Test case 1: AR Extract Loading file level validation



Test case 1 : AR extract loading file level validation

S.no	Description	Expected result	Actual result	Pass/Fail	Root cause	Solution	Client/HRC
1	Format	Tab text delimited	Tab text delimited	Pass	NA	NA	NA
2	Total number of records in File	40000	40000	Pass	NA	NA	NA
3	Number of records inserted into main table	40000	40000	Pass	NA	NA	NA
4	Number of records inserted into child table	40000	37000	Fail	Customer name contains apostrophie and query building fails for that batch of 500 records	Request client to remove apostrophie from customer name.	Client
5	Duplicates	0	1800	Fail	Reference doc number is different for entries with same invoice number and are to be treated as separate entries	Consider reference doc number as part of Primary key	HRC
6	Error	0	200	Fail	Invoice number coming in as blank	To check if invoice number is to be considered as part of primary key	HRC
7	Transformers	Credit debit indicator should have value 'S' or 'H'	Credit debit indicator should have value 'S' or 'H'	Pass	NA	NA	NA

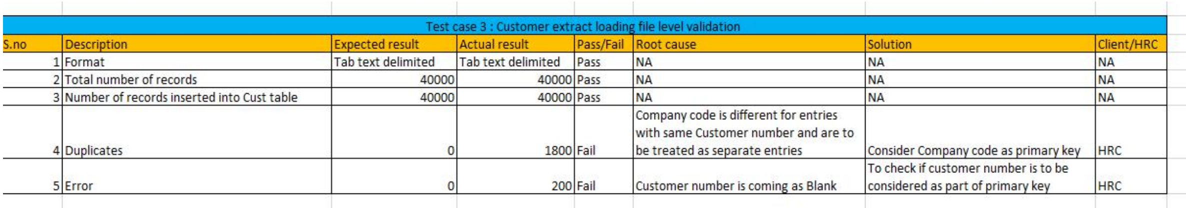
Test case 2: AR Extract Loading data validation



Test case 2: AR extract loading data validation

S.no	Description	Expected result	Actual result	Pass/Fail	Root cause	Solution	Client/HRC
1	Column 1 in file to Customer number	loaded	loaded	Pass	NA	NA	NA
2	Column 2 in file to Assignment ID	failed	failed	Fail	Configure mapping not present	Included column2 to configure mapping	HRC

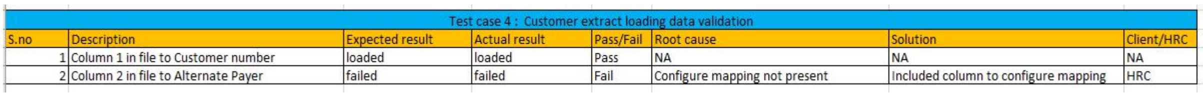
Test case 3: Customer Extract Loading file level validation



Test case 3 : Customer extract loading file level validation

S.no	Description	Expected result	Actual result	Pass/Fail	Root cause	Solution	Client/HRC
1	Format	Tab text delimited	Tab text delimited	Pass	NA	NA	NA
2	Total number of records	40000	40000	Pass	NA	NA	NA
3	Number of records inserted into Cust table	40000	40000	Pass	NA	NA	NA
4	Duplicates	0	1800	Fail	Company code is different for entries with same Customer number and are to be treated as separate entries	Consider Company code as primary key	HRC
5	Error	0	200	Fail	Customer number is coming as Blank	To check if customer number is to be considered as part of primary key	HRC

Test case 4: Customer Extract Loading data validation



Test case 4 : Customer extract loading data validation

S.no	Description	Expected result	Actual result	Pass/Fail	Root cause	Solution	Client/HRC
1	Column 1 in file to Customer number	loaded	loaded	Pass	NA	NA	NA
2	Column 2 in file to Alternate Payer	failed	failed	Fail	Configure mapping not present	Included column to configure mapping	HRC

Figure 15 : Testing Script

CHAPTER : 7

PROJECT PLANNING

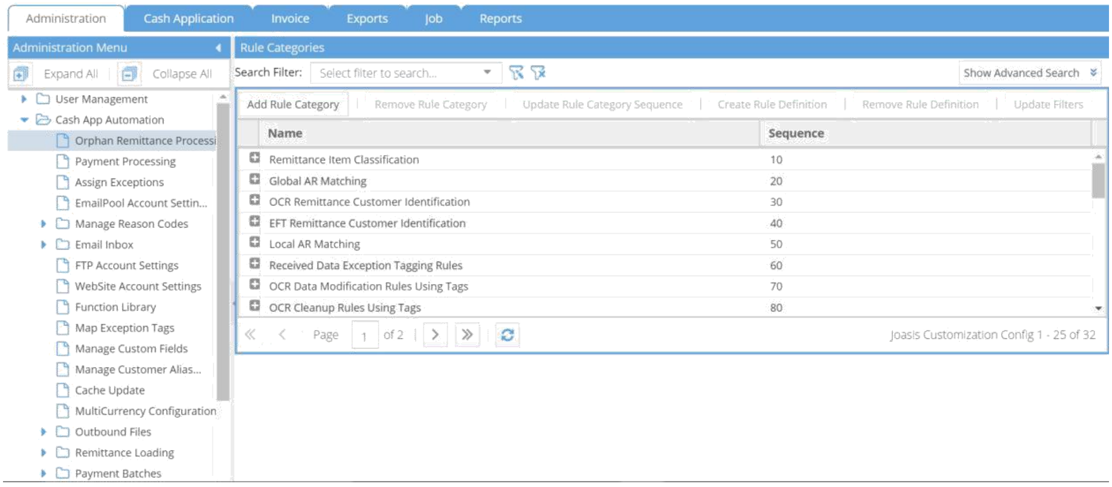
The Project is planned in accordance with the set of rules present in the system.
There are two set of rule sequences:

1. **Orphan Remittance Processing (ORP) rules:** ORP rules operate on the remittances that arrive into our system. These identify the customer on it and also try to link it to the corresponding payment if already available in the system.
2. **Payment Processing (PP) rules:** These rules fire on the payment side and identifies the customer on the payment. The linking of payments to remittance is done from the payment side as well with the help of these rules.

These rules are nothing but pieces of code (java classes) that fire on the payments and remittances once they are appropriately loaded into the correct corresponding tables in the system.

7.1 Orphan Remittance Processing Rules

Orphan Remittance Processing Rules are typically triggered at an Atomic level for a Remittance after the Remittance Items are captured from by the OCR engine and saved in the database. These rules are a sequence of rule categories which are processed in order. Each of these rule categories has a set of rule definitions to choose from. The rule definitions are also executed in the same order of their sequence as we sort them depending on the rule priority. In Fig. 11 the rules categories are presented.



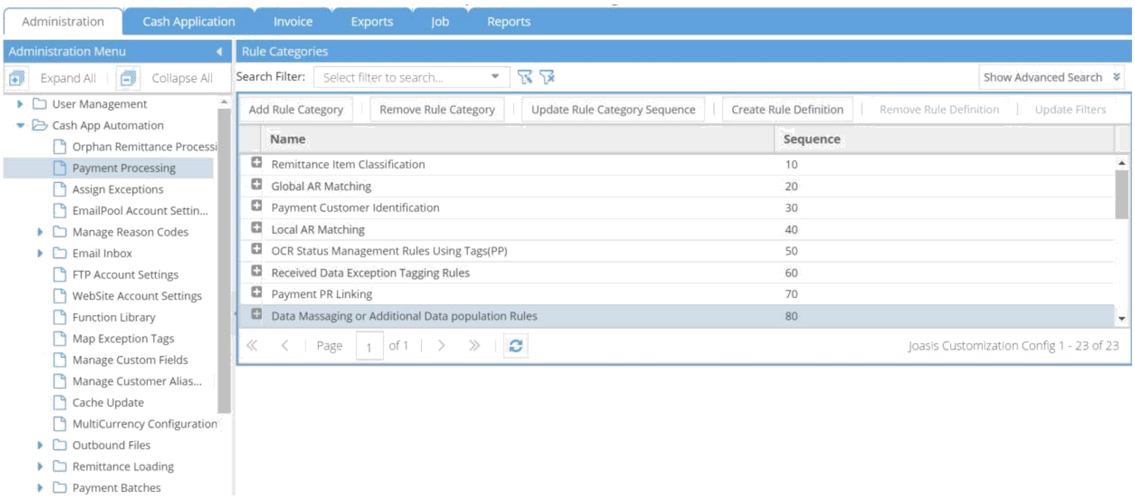
Name	Sequence
Remittance Item Classification	10
Global AR Matching	20
OCR Remittance Customer Identification	30
EFT Remittance Customer Identification	40
Local AR Matching	50
Received Data Exception Tagging Rules	60
OCR Data Modification Rules Using Tags	70
OCR Cleanup Rules Using Tags	80

Joasis Customization Config 1 - 25 of 32

Figure 16 : Screenshot of ORP rules

7.2 Payment Processing Rules

Payment Processing (PP) Rules are typically triggered at an Atomic level for a Payment towards the end after the Payments are loaded in the 'Payment Loading' sub-process. These rules are a sequence of categories, which get processed in a specific order in the database. These rules are a sequence of rule categories which are processed in order. Each of these rule categories has a set of rule definitions to choose from. The rule definitions are also executed in the same order of their sequence as we sort them depending on the rule priority. Following fig shows the list of the Payment Processing Rules.



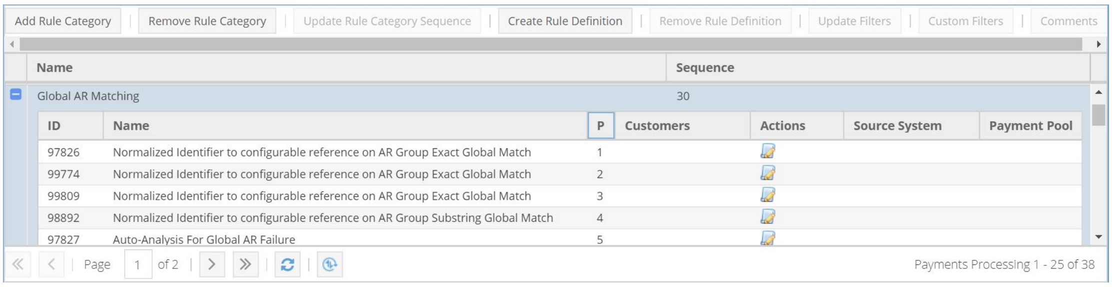
Name	Sequence
Remittance Item Classification	10
Global AR Matching	20
Payment Customer Identification	30
Local AR Matching	40
OCR Status Management Rules Using Tags(PP)	50
Received Data Exception Tagging Rules	60
Payment PR Linking	70
Data Massaging or Additional Data population Rules	80

Figure 17 : Screenshot of PP rules

7.3 Major Rules' Categories in Cash Application

1. **Global AR Matching**

   Global AR Matching Rule objective is to tag the Remittance Items to facilitate the follow-on Customer Identification Rules. At a high level, it does a many-to-many exact match of Identifiers (previously referred to as Reference Numbers) to multiple reference fields in Open AR received from Client's ERP system. Occasionally a substring match is also performed based on Client's situation. But it is recommended to refrain from substring match since it will have a huge performance impact. Global AR Matching is searching across the entire Open AR (Global Lookup) and based on the volume of invoices of the client and the value add of substring match.



Name	Sequence
Global AR Matching	30

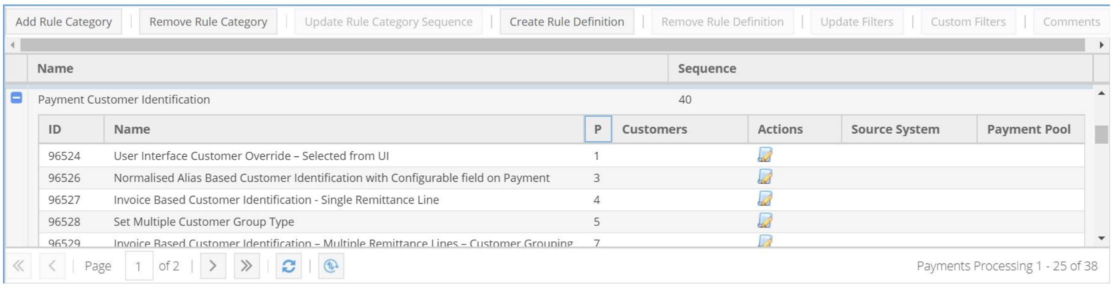
ID	Name	P	Customers	Actions	Source System	Payment Pool
97826	Normalized Identifier to configurable reference on AR Group Exact Global Match	1		✎		
99774	Normalized Identifier to configurable reference on AR Group Exact Global Match	2		✎		
99809	Normalized Identifier to configurable reference on AR Group Exact Global Match	3		✎		
98892	Normalized Identifier to configurable reference on AR Group Substring Global Match	4		✎		
97827	Auto-Analysis For Global AR Failure	5		✎		

Figure 18 : Expansion of a particular rule category (Global AR Matching)

2. Customer Identification Rules

This rule category is a set of rules that are used to identify the customer on a payment or a remittance. The details of customers are present in the Customer

Master data that we extract from the Client’s ERP, this rule matches the customer and their reference numbers mentioned on the payment files. There are various ways based on which customers are identified as illustrated in the following figure.



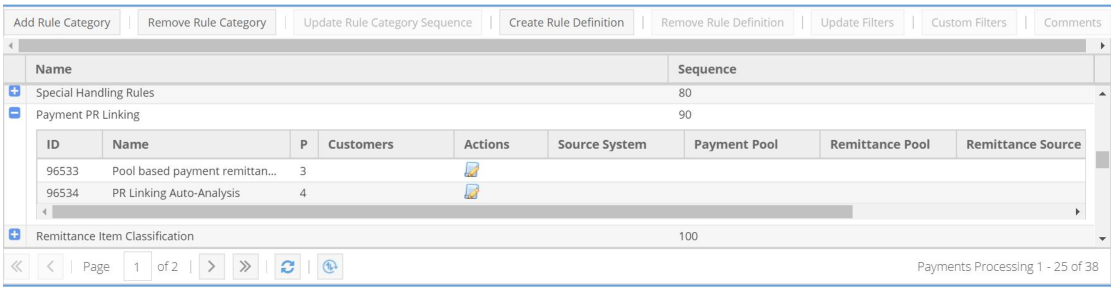
Name	Sequence
Payment Customer Identification	40

ID	Name	P	Customers	Actions	Source System	Payment Pool
96524	User Interface Customer Override – Selected from UI	1		✎		
96526	Normalised Alias Based Customer Identification with Configurable field on Payment	3		✎		
96527	Invoice Based Customer Identification - Single Remittance Line	4		✎		
96528	Set Multiple Customer Group Type	5		✎		
96529	Invoice Based Customer Identification – Multiple Remittance Lines – Customer Grouping	7		✎		

Figure 19 : Expansion of a particular rule category (Customer Identification)

3. Payment Remittance Linking Rules

After finishing the processing of Orphan Remittances via the various ORP Rules, the final step of the Atomic processing is to see if there is a matching Payment in the system. Remittance PR Linking is triggered. Some rule categories below the PR linking category perform an explicit check to see if a payment is linked to a remittance. Only when this is true, the rules in that category will execute. These will be the business rules which generate Transformed Data. Following Fig illustrates the Rule Definition under PR Linking.



Name	Sequence
Special Handling Rules	80
Payment PR Linking	90
Remittance Item Classification	100

ID	Name	P	Customers	Actions	Source System	Payment Pool	Remittance Pool	Remittance Source
96533	Pool based payment remittan...	3		✎				
96534	PR Linking Auto-Analysis	4		✎				

Figure 20 : Expansion of a particular rule category (PR Linking)

7.4 Rules Processing

Linking:

The payments and remittances gathered from various sources are loaded in to the system.

We would link the payments with respective remittances using the payment reference number and/or Customer Number and payment amount. Any other payment remittance linking criteria would be identified (if required) based on the inbound data.

A user Interface button would also be provided in the portal that links payments and remittances when a user clicks it. This would be useful for COMPANY ABC (CLIENT) if at any point in the future, COMPANY ABC (CLIENT) wants to control the linking manually.

Transformation Rules:

Global rules that are applicable to all COMPANY ABC (CLIENT) customers and customer specific rules where required for certain customers will be implemented us transformation rules engine.

We would transform the linked remittance data by applying the rules defined in the rules document.

A user Interface button would also be provided in the portal to apply transformation rules when a user clicks it. This would be useful for COMPANY ABC (CLIENT) if at any point in the future, COMPANY ABC (CLIENT) wants to control the transformation manually.

Output Generation:

Output will be generated the enriched CSV file with the payments data along with the enriched remittance data and place the same in the SFTP server.

There would be as many Output files as the number of Input Payment files received by the System.

A user Interface button would also be provided in the portal to generate the enriched output file when a user clicks it.

CHAPTER : 8

IMPLEMENTATION

Once the ORP and PP rules run on the remittance and payment batches respectively, the corresponding payments and remittances get linked. However, linking of payments to corresponding remittances is essential but not enough to post the payments and clear invoices. The remittance line items identified on the remittance must also be identified in the Open AR extracted from the Client's ERP.

Once the AR matching rule finds the identified remittance line items in the Open AR it marks those invoice line items for clearance. The deductions, discounts, short payments and over payments are also handled by their corresponding rule categories and marked as invoice related or non-invoice related line items.

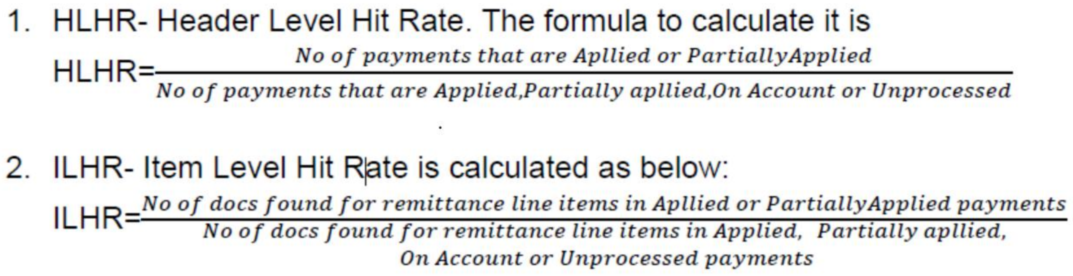
After the entire processing of rules, the ERP simulated status of each payment in the payment batch is generated. ERP simulated status indicates how the payment will get posted in the Client's ERP after the entire processing is completed by the CAA system.

In general, ERP simulated statuses are:

1. **Applied**: All the identified line items are found and sum of all line item amounts is equal to remittance header level amount.
2. **Partially Applied**: At least one document is not found and sum of all line item amounts is equal to remittance header level amount.
3. **On Account**: Customer is identified and sum of all line items amount is not equal to the remittance header level amount.
4. **Unprocessed**: Either corresponding remittance is not found or the Customer is not identified.

A payment is considered **Success** when it is posted as Applied or Partially applied. A payment is not successful when its posting status is On Account or Unprocessed.

The output of an entire project implementation is measured in terms of the following metrics :



1. HLHR- Header Level Hit Rate. The formula to calculate it is

$$HLHR=\frac{No\ of\ payments\ that\ are\ Apllied\ or\ PartiallyApplied}{No\ of\ payments\ that\ are\ Applied,Partially\ apllied,On\ Account\ or\ Unprocessed}$$

2. ILHR- Item Level Hit Rate is calculated as below:

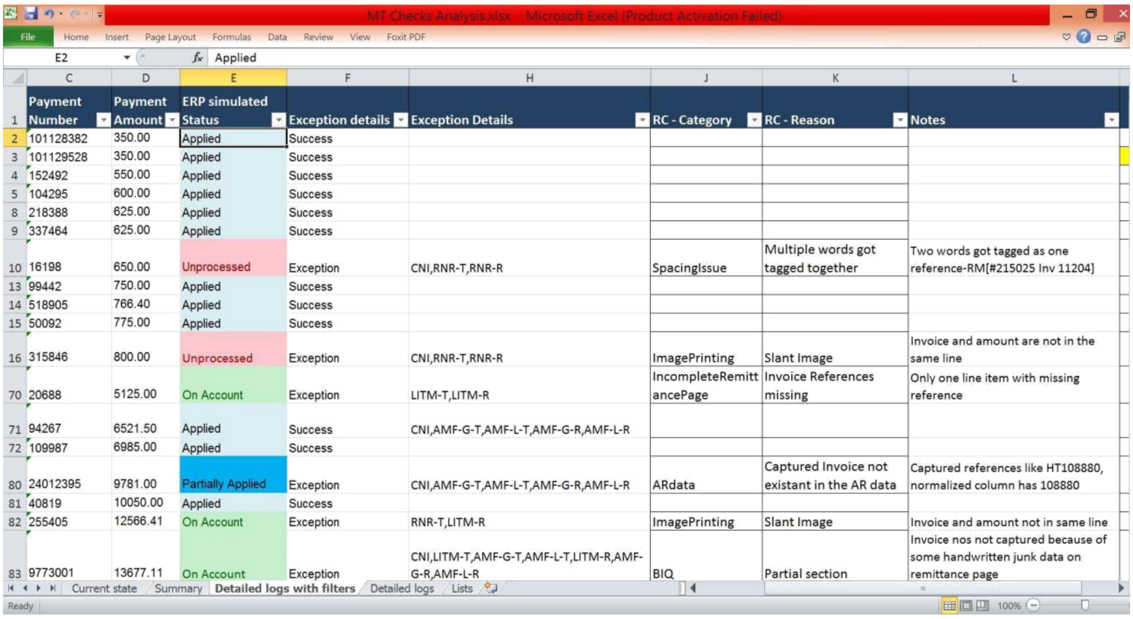
$$ILHR=\frac{No\ of\ docs\ found\ for\ remittance\ line\ items\ in\ Apllied\ or\ PartiallyApplied\ payments}{No\ of\ docs\ found\ for\ remittance\ line\ items\ in\ Applied,\ \ Partially\ apllied,\ On\ Account\ or\ Unprocessed\ payments}$$

These metrics will give us an idea about the percentage or degree of automation the project has been able to achieve and in turn the degree of manual exception handling still required.

8.1 Simulation Phase

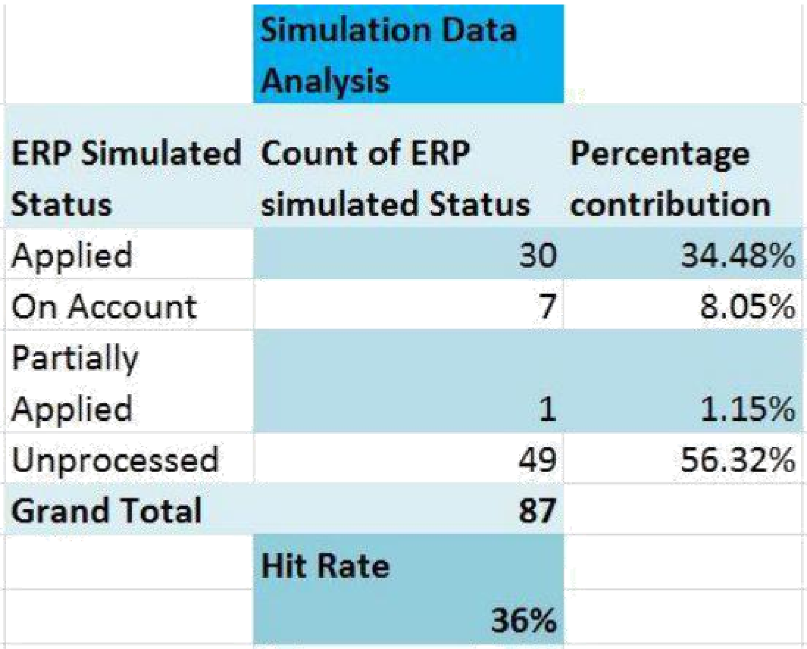
As described in earlier sections, the beginning phases of CAA implementation include Simulation or Testing on a Data set provided by the COMPANY ABC (CLIENT) to understand the degree of automation the CAA technology can achieve using the Gold Rules from the Master framework of CAA. After analysing this, additional rules such as Data Massaging, configuring Regexes, Remittance creation by system etc., depending on the Client's demand, are made in order to deliver a customized software solution for the Client COMPANY ABC.

The simulation analysis includes detailed analysis of every payment in a payment batch as shown in Table 4. The resulting metrics are presented in Table 5 giving the overall Simulation Hit rate.



	C	D	E	F	H	J	K	L
1	Payment Number	Payment Amount	ERP simulated Status	Exception details	Exception Details	RC - Category	RC - Reason	Notes
2	101128382	350.00	Applied	Success				
3	101129528	350.00	Applied	Success				
4	152492	550.00	Applied	Success				
5	104295	600.00	Applied	Success				
8	218388	625.00	Applied	Success				
9	337464	625.00	Applied	Success				
10	16198	650.00	Unprocessed	Exception	CNI,RNR-T,RNR-R	SpacingIssue	Multiple words got tagged together	Two words got tagged as one reference-RM[#215025 Inv 11204]
13	99442	750.00	Applied	Success				
14	518905	766.40	Applied	Success				
15	50092	775.00	Applied	Success				
16	315846	800.00	Unprocessed	Exception	CNI,RNR-T,RNR-R	ImagePrinting	Slant Image	Invoice and amount are not in the same line
70	20688	5125.00	On Account	Exception	LITM-T,LITM-R	IncompleteRemittancePage	Invoice References missing	Only one line item with missing reference
71	94267	6521.50	Applied	Success	CNI,AMF-G-T,AMF-L-T,AMF-G-R,AMF-L-R			
72	109987	6985.00	Applied	Success				
80	24012395	9781.00	Partially Applied	Exception	CNI,AMF-G-T,AMF-L-T,AMF-G-R,AMF-L-R	ARdata	Captured Invoice not existant in the AR data	Captured references like HT108880, normalized column has 108880
81	40819	10050.00	Applied	Success				
82	255405	12566.41	On Account	Exception	RNR-T,LITM-R	ImagePrinting	Slant Image	Invoice and amount not in same line
83	9773001	13677.11	On Account	Exception	CNI,LITM-T,AMF-G-T,AMF-L-T,LITM-R,AMF-G-R,AMF-L-R	BIQ	Partial section	Invoice nos not captured because of some handwritten junk data on remittance page

Table 4 : Simulation Analysis Sheet

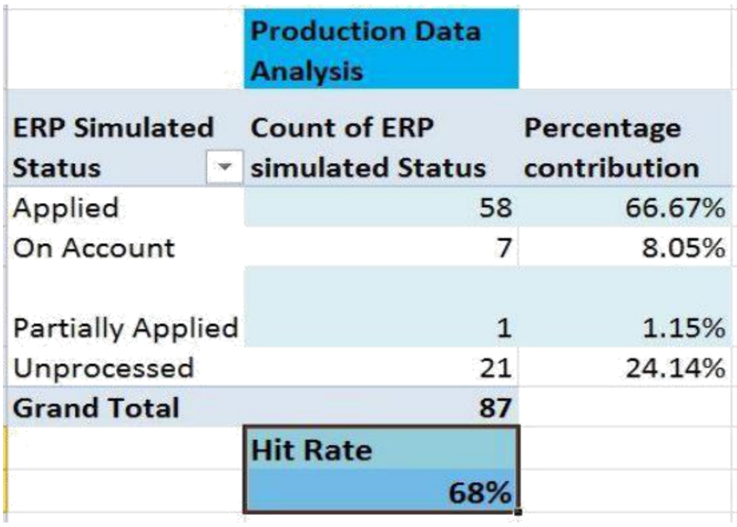


	Simulation Data Analysis	
ERP Simulated Status	Count of ERP simulated Status	Percentage contribution
Applied	30	34.48%
On Account	7	8.05%
Partially Applied	1	1.15%
Unprocessed	49	56.32%
Grand Total	87	
	Hit Rate	
	36%	

Table 5 : Simulation Data Hit Rate (Day 1)

8.2 Production Phase

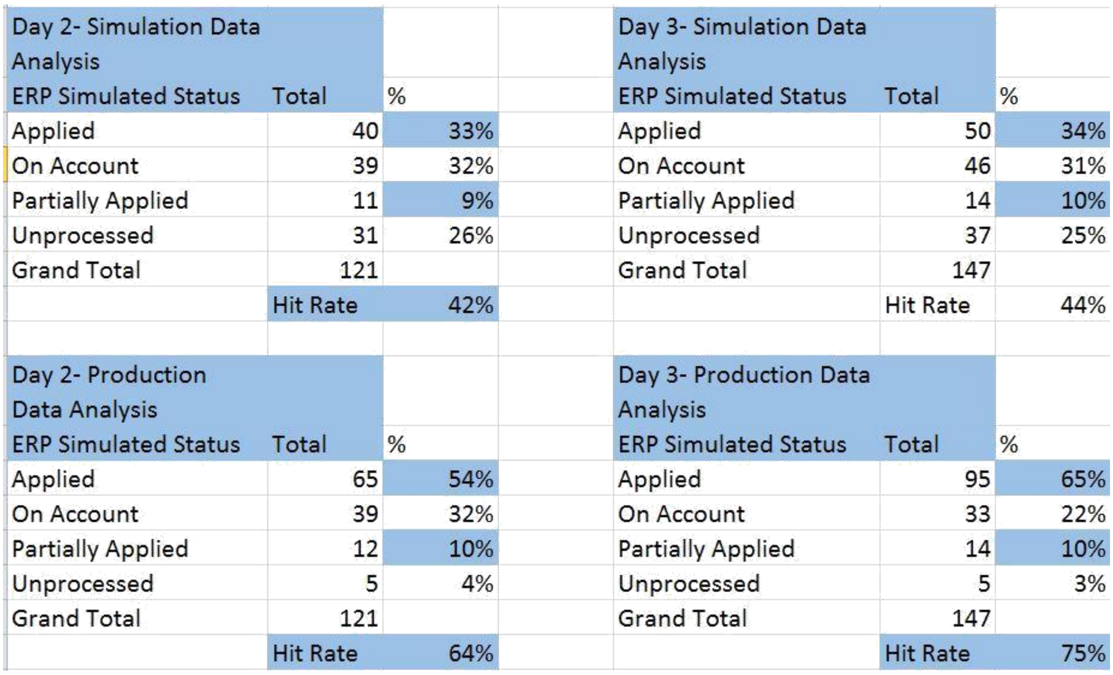
After the Client's specific requirements are gathered during the ASIS sessions, we use various Data massaging rules, Regex configurations to improve the degree of data capture. We also use different discount and deduction coding rules to emulate the business rules that a cash applicant would use to cover the short or over payment scenarios. Using all these rules on the Sample data set, we arrive at a highly improved Hit rate as shown in Table 6. It is the degree of automation that can be delivered to the Client after implementation of the complete project.



	Production Data Analysis	
ERP Simulated Status	Count of ERP simulated Status	Percentage contribution
Applied	58	66.67%
On Account	7	8.05%
Partially Applied	1	1.15%
Unprocessed	21	24.14%
Grand Total	87	
	Hit Rate	
	68%	

Table 6 : Production Data Hit Rate (Day 2)

Analysis of two more batches in Simulation and thereafter in Production phases yields the following results as shown in Table 7.



Day 2- Simulation Data Analysis			Day 3- Simulation Data Analysis		
ERP Simulated Status	Total	%	ERP Simulated Status	Total	%
Applied	40	33%	Applied	50	34%
On Account	39	32%	On Account	46	31%
Partially Applied	11	9%	Partially Applied	14	10%
Unprocessed	31	26%	Unprocessed	37	25%
Grand Total	121		Grand Total	147	
	Hit Rate	42%		Hit Rate	44%
Day 2- Production Data Analysis			Day 3- Production Data Analysis		
ERP Simulated Status	Total	%	ERP Simulated Status	Total	%
Applied	65	54%	Applied	95	65%
On Account	39	32%	On Account	33	22%
Partially Applied	12	10%	Partially Applied	14	10%
Unprocessed	5	4%	Unprocessed	5	3%
Grand Total	121		Grand Total	147	
	Hit Rate	64%		Hit Rate	75%

Table 7 : Simulation and Production Data Hit Rate Comparison (Day 2 and 3)

A graphical representation of the improvement in Hit Rates for the Sample Data of 3 days is shown in Fig. 20

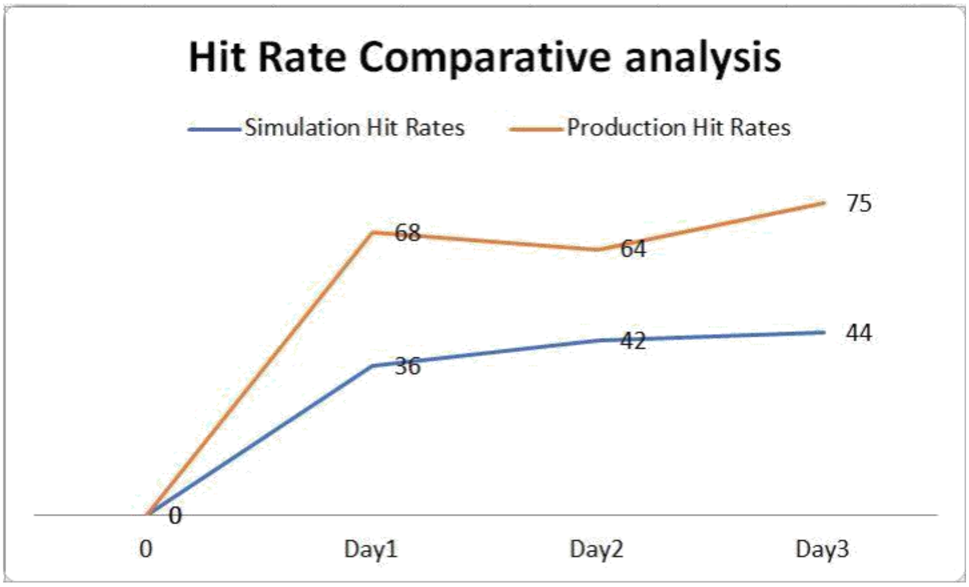


Figure 21 : Day wise comparative analysis of Hit Rates

8.3 Cost Analysis

For a large scale business the average number of human resources required as Full Time Employees (FTE) to apply cash manually in the company's ERP is 5. These FTE's work on a daily basis to clear pending invoices in their system. By providing an automation of rate of 60-70% in production we can reduce their effort by the same percentage.

This would enable the company to employ only 1 FTE thereafter, to handle the exceptions manually but via an extremely user friendly User Interface and a preloaded repository of related remittances.

Cost analysis:

No. of FTEs	**Cost to Company**
5	5x$5,000 = $25,000 per month
Total	**$25,000 per month**

Table 8: ASIS cost analysis before automation

No. of FTEs/Resources	**Cost to Company**
1	1x $5,000=$5,000 per month
CAA end to end implementation and maintenance	$48,000/12 = $4,000 per month
TOTAL	**$9,000 per month**

Table 9: TO BE cost analysis after CAA automation

Since RDC machine is a one-time investment, thus, on a per day basis the effective profit gained by the Client Company ABC on using CAA is **$16,000.**

CHAPTER : 9

CONCLUSION AND FUTURE SCOPE

An automated cash application process goes through the same procedure as the manual cash application system but is able to match payments and remittances at a much faster speed. As the cash application process has grown more and more complex, many companies have moved to an automated process. Thereby reducing the staff work load, cost and work burnout.

Another important feature of automated cash application solution is that it can match payment, remittance, and invoice information in one system and simulate the ERP posting status before the cash is actually applied in the client's ERP. This feature highly optimizes the process of exception handling on the Client's side. The analysis of the posting status of each payment in a payment batch helps the consultant to further optimize the solution in the testing phase before delivering the final end product to the Client. In the production phase, this can further be used to debug issues before they are actually raised from the Client's side.

As the Cash App Automation develops, it will eliminate most of the work of valuable human resources at the Client's end, avoiding errors, freeing up resources, and reducing costs. With the right technology in place, businesses can resolve any challenges they face in their time-sensitive cash application process.

In the entire Order to Cash flow, business processes can be extremely simplified by using such automated systems in place of valuable human resources for more innovative and useful purposes. Not only the Cash Application system, but also other manual systems in place, such as Credit management, Invoice presentment and payment, Collections management as well as the Discounts and Dispute settlement systems can be highly automated and synchronized with each other to deliver a complete solution for managing the Cash flow in the Accounts Receivables section of the Financial Supply Chain Management (FSCM) for every business.

The use of RDC functionality can reduce the companies' dependencies on Banks. Also, the exception handling feature can help in correcting capture mistakes if any. This can lead to complete independence and standardization of applying cash.

REFERENCES

[1] F. Liu, "Cash flow management of modern enterprises under the guidance of the application of internal control," *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, Deng Leng, 2011, pp. 6176-6178, China

[2] M. He, C. Ren, B. Shao, Q. Wang and J. Dong, "Financial supply chain management," *Proceedings of 2010 IEEE International Conference on Service Operations and Logistics, and Informatics*, Qingdao, Shandong, 2010, pp. 70-75.

[3] T. Li and W. Wang, "The Study on Application Value of Cash Flow Information Connotation in Financial Early-Warning," *2010 International Conference on Management and Service Science*, Wuhan, 2010, pp. 1-4.

[4] J. H. Chen and W. H. Chen, "Factoring Account Receivables Towards Mitigating Cash Flow Fluctuation for Construction Projects," *2008 IEEE International Conference on Communications*, Beijing, 2008, pp. 5538-5542.

[5] D. D. Wu, D. L. Olson and C. Luo, "A Decision Support Approach for Accounts Receivable Risk Management," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 12, pp. 1624-1632, Dec. 2014.