

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018



”LOYALTY ENGINE”

This is submitted in partial fulfillment of the curriculum prescribed
for the award of the degree of Bachelor of Engineering in
Computer Science & Engineering by

1CR14CS090 Karteek Reddy P V

Under the Guidance of

Mrs. Priya L
Assistant Professor
Department of CSE, CMRIT, Bengaluru



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
#132, AECS LAYOUT, IT PARK ROAD, BENGALURU - 560037

2017-18

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018



Certificate

This is to certify that the project entitled "LOYALTY ENGINE" is a bonafide work carried out by **KartEEK Reddy P V** in partial fulfillment of the award of the degree of Bachelor of Engineering in **Computer Science & Engineering** of Visvesvaraya Technological University, Belgaum, during the year 2017-18. It is certified that all corrections / suggestions indicated during reviews have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of Guide
Mrs. Priya L
Assistant Professor
Department of CSE
CMRIT, Bengaluru - 37

Signature of HoD
Dr. Jhansi Rani P
Professor & Head
Department of CSE
CMRIT, Bengaluru - 37

Signature of Principal
Dr. Sanjay Jain
Principal
CMRIT,
Bengaluru - 37

External Viva

Name of the Examiners	Institution	Signature with Date
1. -----	-----	-----
2. -----	-----	-----

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018



Declaration

I **Mr. Karteek Reddy P V**, bonafide students of CMR Institute of Technology, Bangalore, hereby declare that the dissertation entitled "**LOYALTY ENGINE**" has been carried out by me under the guidance of **Mrs. Priya L**, Assistant Professor of CSE Department, CMRIT, Bangalore, in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in **Computer Science Engineering**, of the Visvesvaraya Technological University, Belgaum during the academic year 2017-2018. The work done in this dissertation report is original and it has not been submitted for any other degree in any University.

Karteek Reddy P V

Acknowledgement

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without mentioning people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So,with gratitude we acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

I would like to thank **Dr. Sanjay Jain**, Principal, CMRIT, Bangalore for providing excellent academic environment in the college and his never-ending support for the B.E program.

I would like to thank **Dr. Jhansi Rani**, Professor & HOD and **Mrs.Swathi Y**,Associate Professor & Program Coordinator, Department of Computer Science, CMRIT, Bangalore who shared their opinions and experiences through which I received the required information crucial for the project.

I would also like to thank **Mr. Sudhakar K N** , Associate Professor & Project Coordinator, Department of Computer Science, CMRIT, Bangalore who shared his opinions and experiences through which I received the required information crucial for the project.

I consider it a privilege and honour to express my sincere gratitude to my internal guide **Mrs. Priya L** , Assistant Professor, Department of Computer Science & Engineering, for his valuable guidance throughout the tenure of this project work.

I would also like to thank all the faculty members who have always been very co-operative and generous. Conclusively, I also thank all the non-teaching staff and all others who have done immense help directly or indirectly during my project

Karteek Reddy P V

Table of Contents

Table of Contents	ii
List of Figures	iv
List of Tables	vi
Abstract	vii
1 PREAMBLE	1
1.1 Introduction	1
1.2 Existing System	2
1.3 Proposed System	2
1.4 Organization of the project report	3
2 LITERATURE SURVEY	5
3 THEORETICAL BACKGROUND	9
3.1 Java Spring	9
4 SYSTEM REQUIREMENT SPECIFICATION	14
4.1 Functional Requirements	14
4.2 Non-Functional Requirements	14
4.3 Hardware Requirements	18
4.4 Software Requirements	18
5 SYSTEM ANALYSIS	19
5.1 Overview	19
5.2 Feasibility Study	19
6 SYSTEM DESIGN	21
6.1 System development methodology	21
7 IMPLEMENTATION	26

7.1	Loyalty-Issuer Portal	26
7.2	Loyalty-Admin Portal	32
7.3	Loyalty-Customer Portal	42
8	TESTING	47
8.1	Testing Methodologies	47
8.2	Unit Testing	49
8.3	Integration Testing	49
8.4	System Testing	51
8.5	Quality Assurance	52
9	RESULTS & EXECUTION	55
10	CONCLUSION & FUTURE ENHANCEMENT	57
10.1	Conclusion	57
10.2	Future Enhancement	57
	REFERENCES	58

List of Figures

6.1	Prototyping model	22
6.2	Architecture of the entities	23
7.1	Loyalty-Issuer Portal Flow Diagram	26
7.2	Login page	27
7.3	Dash Board	27
7.4	Creation of Program Manager	28
7.5	Creation of Card Issuer	29
7.6	Creation of a Message	30
7.7	Creation of a Campaign	31
7.8	Creating a Reward Catalog	31
7.9	Creation of Loyalty Program	32
7.10	Loyalty-Admin Flow Diagram	33
7.11	Login Page of Loyalty-Admin Portal	33
7.12	Dashboard of Loyalty-Admin Portal	34
7.13	Creation of a bank	35
7.14	Creation of BIN	35
7.15	Creation of the Program Manager	36
7.16	Creation of a Partner	37
7.17	Creation of a Sales Person	38
7.18	Configuration of the loyalty program	39
7.19	Creation of the Account Program	39
7.20	Creation of the Fee Program	40
7.21	Creation of Card Program	41
7.22	Mapping the Entities	42
7.23	Details of the customer	42
7.24	Identity of the customer	43
7.25	Email generated to the customer	43
7.26	Loyalty-Customer portal Login page	44
7.27	Dashboard of the Loyalty-Customer portal	44

7.28	Loyalty points Display	45
7.29	Catalog to redeem the loyalty points	45
7.30	Conversion of loyalty points in to money	46
7.31	Transaction history	46
9.1	Redemption of loyalty points using the Reward Catalog	55
9.2	Redemption of loyalty points using the top-up cash	56
9.3	Transaction history	56

List of Tables

1.1	Phase Description	3
8.1	Functions for Input	49
8.2	Functions for Output	50
8.3	Integration Testing	50

Abstract

This document gives high level functional requirements of the Loyalty Engine. Loyalty Engine offers Loyalty solutions to card issuers and Merchants. Loyalty engine should be a multitenant solution and should support multiple loyalty programs defined by different card issuers and different merchants. The platform should support creation and management of multiple loyalty programs seamlessly. In addition to card issuers and Merchants, the platform has Program managers who will own and manage the Loyalty Programs. A program manager could be a card issuer or a merchant or a separate third-party loyalty solution provider. In Loyalty Solution, customers are mapped to a loyalty program and they belong to the program manager.

Keywords: Program Manager, Merchant, Card Issuer, Customer.

Chapter 1

PREAMBLE

1.1 Introduction

Loyalty programs are structured marketing strategies designed by merchants to encourage customers to continue to shop at or use the services of businesses associated with each program. These programs exist covering most types of commerce, each one having varying features and rewards-schemes.

In marketing generally and in retailing more specifically, a loyalty card, rewards card, points card, advantage card, or club card is a plastic or paper card, visually similar to a credit card, debit card, or digital card that identifies the card holder as a participant in a loyalty program. Loyalty cards (both physical and digital) relate to the loyalty business-model. In the United Kingdom such a card is typically called a "loyalty card", in Canada a "rewards card" or a "points card", in the United States of America a "discount card", a "club card" or a "rewards card" and in Australia a "customer card" or "brand-name-in-question" card, for example: a "Coles Card". Cards typically have a bar-code, mag-stripe or RFID chip that can be easily scanned, although some are chip cards or proximity cards. Small key-ring cards (also known as key-tags) which serve as key fobs bring convenience in carrying and ease of access.

By presenting such a card, purchasers typically earn the right either to a discount on the current purchase, or to an allotment of points that they can use for future purchases. Hence the card is the visible means of implementing a type of what economists call a two-part tariff. Application forms for cards usually entail agreements by the store concerning customer privacy, typically non-disclosure (by the store) of non-aggregate data about customers. The store uses aggregate data internally (and sometimes externally) as part of its marketing research. Over time the data can reveal, for example, a given customer's favorite brand of beer, or whether he or she

is a vegetarian. Where a customer has provided sufficient identifying information, the loyalty card may also be used to access such information to expedite verification during receipt of cheques or dispensing medical prescription preparations, or for other membership privileges.

1.2 Existing System

The Loyalty cards that are issued by the merchant are limited to that particular store. So we cannot use these cards in other stores or super markets. For some stores and super markets the cards issued by them have the limited validity time.

The conversion ratio is very low. Conversion ratio is the ratio of the amount to the number of loyalty points consumed by that amount. There is no particular dedicated program manager for the loyalty program so, the issues raised by the customer cannot be resolved by the card issuer as the loyalty programs are implemented by the external agents. Due to this the creation and maintenance of the loyalty program will be difficult.

1.3 Proposed System

The proposed system has the following advantages on the existing system.

- Cards that are issued are connected directly to the Debit or Credit card of the customer.
- The customers can convert the loyalty points in to money and can transfer the money in to the account that the card is linked with.
- Customers can use the loyalty points in any store by converting them in to money.
- Creation and maintenance of the loyalty program is easy as there is a dedicated program manager.
- The issues that are occurred can be resolved easily by the card issuer as he is connected with the program manager.
- The validity of the card will be lifetime and the customer can redeem his or her loyalty points either by picking some items from the catalog or convert the points in to top-up.
- Loyalty points conversion ratio is very high and can be decided by the card issuer

himself.

Table 1.1: Phase Description

Phase	Task	Description
Phase1	Analysis	Analyzing the core of the IEEE paper and provide Literature review based on analysis.
Phase2	Literature survey	Collect raw data and elaborate on literature surveys.
Phase3	System analysis	Analyses the requirements of the project and lists the specific requirements needed.
Phase4	Design	Object designing and Functional description .
Phase5	Implementation	Implement the code based on the object specification .
Phase6	Testing	Test the project according to Test Specification .
Phase7	Documentation	Prepare the document for this project with conclusion and future enhancement.

1.4 Organization of the project report

The project report is organized as follows:

Chapter 2: Literature Review - Gives a brief overview of the survey papers and the research sources that have been studied to establish a thorough understanding of the project under consideration.

Chapter 3: Theoretical Background - Establishes groundwork for the proposed project by giving a detailed analysis of the project topic, existing research relevant to the project, arguments in favor and against the existing solutions and finally explores the motivation behind the proposed solution.

Chapter 4: System Requirement Specification - Discusses in details about the different kinds of requirements needed to successfully complete the project.

Chapter 5: System Analysis - gives details about several analysis that are performed to facilitate taking decision of whether the project is feasible enough or not.

Chapter 6: System Design - Gives the design description of the project, conceptual and detailed design well supported with design diagrams.

Chapter 7: Implementation - Discusses the implementation details of the project and reasons the use of the programming language and development environment.

Chapter 8: Testing - Briefs the testing methods used for testing the different modules in the project.

Chapter 9: Results and Performance Analysis - Gives the snapshots and graphs of the proposed protocols.

Chapter 10: Conclusion and Future Scope - Gives the concluding remarks of the project, throwing light on its future aspects.

Chapter 2

LITERATURE SURVEY

Literature survey is mainly carried out in order to analyze the background of the current project which helps to find out flaws in the existing system and guides on which unsolved problems we can work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated to propose solutions and work on this project.

[1] Su Lujun. Study on the Mechanism of Service Fairness Influencing to Consumer Loyalty. Journal of Shanxi Finance and Economics University.2010, 07, pp.94-103.

The direct expression of customer loyalty is that customers reject the temptation of price-cutting by other enterprises, frequently purchase the products of the enterprises they trust and even volunteer to introduce the products or enterprises they trust to their families and friends. The competition of retail enterprises in the market is in fact to fight for customers. In the competitive market, anyone who has more customers will take the preemptive opportunities and then dominate the market. One important index weighing the market position and core-competitiveness of retail enterprises is customer loyalty.

[2] Zou Jianlan.Customer Loyalty Cultivation Based On Customer Satisfaction [J].Journal of Harbin University of Commerceh, 2010, 04, pp.58-60

Customer satisfaction refers to the feeling state of cheer or disappointment formed after customers compare the actual effect they've felt after purchasing the products and service with their expected value. Maintaining and improving the customer loyalty of retail enterprises will not only create a growing marketing achievement and

reduce the operating cost of enterprises but also bring more customers through consumers' word-of-mouth propaganda and the demonstration effect of their repetitive purchasing, thus realizing the long-term development of the retail enterprises. Therefore, it is very important for retail enterprises to improve customer loyalty by all manner of means.

[3] Customer Loyalty and Customer Relationship Management Pengwei Zhang, Min Li, Xiaojing Jiao, and Ruijin Zhou Henan Institute of Science and Technology Xinxiang, China.

The contemporary company attaches great importance to marketing relationship and customer relations is the core of this relationship. Further, customer satisfaction and loyalty is the core of the customer relationship management. Sometimes, high customer satisfaction causes low profit because enterprises do not realize that strengthening the loyalty of the aimed customer is the key of customer relationship management.

Contemporarily, Enterprises attach great importance to marketing relationship, which is a looked marketing activity as the process that a company interacts with customers, suppliers, distributors, competitors, government agencies and other publics. Its key is the establishment and development of good public relationship. Marketing relationship not only focuses on developing and maintaining relations with his clients but also expands the horizons of marketing. It involves all of the relationship between the enterprises and its stakeholders. Even so, among these relationships, the core is still between enterprises and customer. Wal-mart and Carrefour stores also attach great importance to the relations with suppliers, but comparably with customer relationship, supplier relationships aim is to service customer relationship. The procurement system of Wal-mart and Carrefour is extremely harsh, which is the efforts of maintaining long and good relationship with customers.

The key of customer relationship is the commercial relationship, not the personal relationship, and personal relationship serves the commercial relationship. Here, we have to distinguish the difference between customers and friends. Customers are business concepts. Good customers must have the demand, have decision-making power and have the ability to pay, and satisfied with the company. However, friend is an emotional concept, which refers to everyone has common interests, hobbies, and cares and helps each other, etc. When doing customer relationship marketing, managers must always clearly realize what we uphold are customer relationships, not friend relationships. Although both can be gained at the same time, if not, managers must keep

reminding themselves that customer relationship is the first, then friend relationship. Just as a business saying, some people can be friends without doing business, some people are the opposite.

[4] Customer Loyalty Attributes: A Perspective Alok Kumar Rai 1 Medha Srivastava.

Many academicians have accepted the significance of loyalty in service industries (Bloemer et al., 1999; Caruana, 2002; Asuncion et al., 2004) and its potential impact on the development of sustainable competitive edge (Keaveney, 1995; Gremler and Brown, 1996) for the service firms. This may be attributed to the unique nature of services, increased dependency on technology and greater customer involvement in service delivery. A base of loyal customers can do wonder in terms of economic rewards and new business prospects as winning a new customer can cost as much as 6 times more than the cost of retaining an old one (Rosenberg and Czepiel, 1984) whereas profits can be increased from 25% to 125% if the potential migration is decreased by 5 percent depending upon the particular industry (Reichheld and Sasser, 1990). Customer loyalty clearly brings in significant benefits to the business and calls for a deeper investigation into the factors that act as its originator and contribute in its enhancement since, as mentioned by Johnson, Herrmann, and Huber (2006), the antecedents of customer loyalty are convoluted and dynamic, changing and evolving over time. The purpose of this paper is to discuss customer loyalty and its significance in the modern business arena through a comprehensive survey of literature. Further, it seeks to explore various factors that serve as antecedents to customer loyalty development. In other words, investigating the customers' checklist that they refer before deciding to sustain and develop their relationship with the company and its offerings is the key issue that this paper aims to address. The study intends to provide a framework for development of a scale for assessing customer loyalty and identifying what a particular class of loyalty would result into in terms of its contribution to the organization. The above identified framework would prove to be of great significance for service organizations in appraising different loyalty programs and also in segmenting customers depending upon organizational requirements.

Globalized markets and borderless flow of information have resulted in intense competitive pressures and increased customer expectations. Productivity, quality, customer satisfaction are the buzz-words in today's business scenario that demand considerable efforts on the part of the company. Further, to attain the basic business goals of survival and growth, businesses are looking for ways to attract and retain

customers in the long run. It is established now that every business needs to understand and meet customers' expectations in order to strive and thrive in the market. Customers have become the focal point of almost all the businesses now and thus, deserve all the attention and importance. However, due to heightened expectations, escalated competition and rapid ingress of new business concepts and formats, companies are finding it increasingly difficult to retain their customers along with managing to be profitable. Instability of the economic environment in recent times has also contributed to the loyalty issues in businesses. In order to develop and sustain loyalty among the customers, it is important to find out what drives loyalty in a particular market. The factors which lead to loyalty need to be uncovered and understood before designing and implementing the strategies for customer retention and loyalty. Literature proposes relationships between customer loyalty and various other business constructs such as quality, satisfaction, trust and so on. These relationships need to be investigated and understanding the effect of these constructs on loyalty will surely provide an insight into customer loyalty formation.

Services have been widely researched and analyzed for their unique characteristics and intensive customer orientation. Some of the major concepts that have been studied to explore the true nature of services are Service Quality (Parasuraman, Zeithaml & Berry, 1985; Gronroos, 1988; Cronin & Taylor, 1992; O'Neill, 1992; Oliver, 1997), Satisfaction (Oliver, 1993 & 1997; Wirtz & Bateson, 1999; Zeithaml & Bitner, 2000), Loyalty (Dick & Basu, 1994; Oliver, 1997; Bowen & Shoemaker, 1998; Reichheld & Sasser, 1990; Heskett, Sasser & Schlesinger, 1994; McMullan & Gilmore, 2003; McMullan, 2005) and Complaint Management Systems (Boshoff, 1997 & 1999; Mattila, 2001; Boshoff & Staude, 2003; Craighead, Karwan, & Miller, 2004; Mattila & Patterson, 2004). Role of customer loyalty gains more prominence when applied in the context of services due to the higher human involvement in comparison to goods. Such people intrinsic character of services along with its intangible and perishable nature, enhance the scope for error at the time of service delivery and amplify the role and significance of human relationships in business transactions. Also, heightened competition has resulted in fewer possibilities for differentiation. To tackle the competitive pressures and gain an edge in the market, companies are now looking forward to leverage upon the intangible nature of services and the significant human interface involved there. Customer loyalty has been included in the strategic objectives of many companies due to the competitive strength it offers.

Chapter 3

THEORETICAL BACKGROUND

Theoretical background highlighting some topics related to project work. The description contains several topics which are worth to discuss and also highlight some of their limitation that encourage going on finding solution as well as highlights some of their advantages for which reason these topics and their features are used in this project.

3.1 Java Spring

The Spring Framework is an application framework and invention of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to, or even replacement for the Enterprise java bean (EJB) model.

The first version was written by Rod Johnson, who released the framework with the publication of his book *Expert One-on-One J2EE Design and Development* in October 2002. The framework was first released under the Apache 2.0 license in June 2003. The first milestone release, 1.0, was released in March 2004 with further milestone releases in September 2004 and March 2005. The Spring 1.2.6 framework won a Jolt Production Award and a JAX innovation award in 2006. Spring 2.0 was released in October 2006, Spring 2.5 in November 2007, Spring 3.0 in December 2009, Spring 3.1 in December 2011, and Spring 3.2.5 in November 2013. Spring Framework 4.0 was released in December 2013.[5] Notable improvements in Spring 4.0 included support for Java SE (Standard Edition) 8, Groovy 2, some aspects of Java EE 7, and WebSocket. Spring Framework 4.2.0 was released on 31 July 2015 and was immediately

upgraded to version 4.2.1, which was released on 01 Sept 2015. It is "compatible with Java 6, 7 and 8, with a focus on core refinements and modern web capabilities". Spring Framework 4.3 has been released on 10 June 2016 and will be supported until 2020. It "will be the final generation within the general Spring 4 system requirements .

3.1.1 Modules of Spring

The Spring Framework includes several modules that provide a range of services:

- Spring Core Container: this is the base module of Spring and provides spring containers (BeanFactory and ApplicationContext).
- Aspect-Oriented Programming: enables implementing cross-cutting concerns.
- Authentication and authorization: configurable security processes that support a range of standards, protocols, tools and practices via the Spring security sub-project (formerly Acegi Security System for Spring).
- Convention over Configuration: a rapid application development solution for Spring-based enterprise applications is offered in the Spring Roo module.
- Data access: working with relational database management systems on the Java platform using java Database connectivity (JDBC) and object-relational mapping tools and with NoSQL databases.
- Inversion of control container: configuration of application components and lifecycle management of Java objects, done mainly via dependency injection.
- Messaging: configurative registration of message listener objects for transparent message-consumption from message queues via Java Message Service (JMS), improvement of message sending over standard JMS APIs.
- Model-View-Controller: an HTTP - and servlet -based framework providing hooks for extension and customization for web applications and RESTful (representational state transfer) Web services.
- Remote access framework: configurative remote procedure call (RPC)-style marshalling of Java objects over networks supporting java remote method invocation (RMI), CORBA (Common Object Request Broker Architecture) and HTTP -based protocols including Web Services ((SOAP)Simple Object Access Protocol).
- Transaction Management: unifies several transaction management APIs and coordinates transactions for Java objects.
- Remote management: configurative exposure and management of Java objects for local or remote configuration via Java Management Extension (JMX).

3.1.2 Model View Controller

The Spring Framework features its own Model-View-Controller (MVC) web application framework, which wasn't originally planned. The Spring developers decided to write their own Web framework as a reaction to what they perceived as the poor design of the (then) popular Jakarta Struts Web framework, as well as deficiencies in other available frameworks. In particular, they felt there was insufficient separation between the presentation and request handling layers, and between the request handling layer and the model.

Like Struts, Spring MVC is a request-based framework. The framework defines strategy interfaces for all of the responsibilities that must be handled by a modern request-based framework. The goal of each interface is to be simple and clear so that it's easy for Spring MVC users to write their own implementations, if they so choose. MVC paves the way for cleaner front-end code. All interfaces are tightly coupled to the Servlet API. This tight coupling to the Servlet API is seen by some as a failure on the part of the Spring developers to offer a high-level abstraction for Web-based applications. However, this coupling makes sure that the features of the Servlet API remain available to developers while offering a high abstraction framework to ease working with said API.

The Dispatcher Servlet class is the Front-Controller of the framework and is responsible for delegating control to the various interfaces during the execution phases of an HTTP request.

The most important interfaces defined by Spring MVC, and their responsibilities, are listed below:

- Controller: comes between Model and View to manage incoming requests and redirect to proper response. It acts as a gate that directs the incoming information. It switches between going into model or view.
- Handler Adapter: execution of objects that handle incoming requests.
- Handler Interceptor: interception of incoming requests comparable, but not equal to Servlet filters (use is optional and not controlled by Dispatcher Servlet).
- Handler Mapping: selecting objects that handle incoming requests (handlers) based on any attribute or condition internal or external to those requests.
- Locale Resolver: resolving and optionally saving of the locale of an individual user.
- Multipart Resolver: facilitate working with file uploads by wrapping incoming requests.
- View: responsible for returning a response to the client. Some requests may go

straight to view without going to the model part; others may go through all three.

→ View Resolver: selecting a View based on a logical name for the view (use is not strictly required)

Each strategy interface above has an important responsibility in the overall framework. The abstractions offered by these interfaces are powerful, so to allow for a set of variations in their implementations, Spring MVC ships with implementations of all these interfaces and together offers a feature set on top of the Servlet API. However, developers and vendors are free to write other implementations. Spring MVC uses the Java `java.util.Map` interface as a data-oriented abstraction for the Model where keys are expected to be string values.

The ease of testing the implementations of these interfaces seems one important advantage of the high level of abstraction offered by Spring MVC. Dispatcher Servlet is tightly coupled to the Spring inversion of control container for configuring the web layers of applications. However, web applications can use other parts of the Spring Framework including the container and choose not to use Spring MVC.

3.1.3 Spring JDBC

Spring JDBC is an abstraction framework for JDBC that provides easier access to data sources without all the exception handling and parsing of SQL fetch results. Spring JDBC basically does lots of things for you.

3.1.3.1 Advantages of Spring JDBC

→ `JdbcDaoSupport` - Convenient super class for JDBC data access objects, providing a `JdbcTemplate` based on it to subclasses.

→ Spring provides an abstract exception layer, moving verbose and error-prone exception handling out of application code into the framework. The framework takes care of all exception handling; application code can concentrate on extracting results by using appropriate SQL.

→ Spring provides a significant exception hierarchy for your application code to work with in place of `SQLException`.

→ The Spring framework provides the `org.springframework.jdbc.support.nativejdbc.NativeJdbcExtractor` interface and some implementations (such as `SimpleNativeJdbcExtractor`) of this interface. These are useful for accessing Oracle features via an Oracle connection or `ResultSet` when the

connection is "wrapped" by another DataSource (such as that used with some application servers) or obtained through certain connection pools.

→ For creating instances of `oracle.sql.BLOB` (binary large object) and `oracle.sql.CLOB` (character large object), Spring provides the class.

`org.springframework.jdbc.support.lob.OracleLobHandler`.

→ The Spring-provided `OracleSequenceMaxValueIncrementer` class provides the next value of an Oracle sequence. It effectively provides the same information that would be provided if you used the following command directly: `select someSequence.nextval from dual` (where `someSequence` is the name of your sequence in the Oracle database).

→ An advantage of this approach is that the `DataFieldMaxValueIncrementer` interface can be used in a DAO hierarchy without tight coupling of the Oracle-specific implementation.

Chapter 4

SYSTEM REQUIREMENT SPECIFICATION

Software requirement Specification is a fundamental document, which forms the foundation of the software development process. It not only lists the requirements of a system but also has a description of its major feature.

4.1 Functional Requirements

Functional Requirement defines a function of a system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements: -

- Input test case must not have compilation and runtime errors.
- The system must not stop working when kept running for even a long time.
- The system must function as expected for every set of test cases provided.
- The system should generate the output for given input test case and input parameters.
- The system should generate on-demand services.

4.2 Non-Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and

store occupancy. Nonfunctional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:-

- Product Requirements
- Organizational Requirements
- User Requirements
- Basic Operational Requirements

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing nonfunctional requirements is detailed in the system architecture. Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be. Functional requirements are usually in the form of system shall do , an individual action of part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of system shall be , an overall property of the system as a whole or of a particular aspect and not a specific function. The systems' overall properties commonly mark the difference between whether the development project has succeeded or failed.

The non-functional requirements of our project are:-

- Response time The time the system takes to load and the time for responses on any action the user does.
- Processing time - How long is acceptable to perform key functions or export / import data?
- Throughput The number of operations the system needs to handle must be kept in mind.
- Storage - The amount of data to be stored for the system to function.
- Locations of operation - Geographic location, connection requirements and the restrictions of a local network prevail.
- Architectural Standards The standards needed for the system to work and sustain.

4.2.1 Product Requirements

→ **Portability:** Since the SLR system is designed to run using OpenCV (whose library is written in C), the system is portable.

→ **Correctness:** It follows a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data.

→ **Ease of Use:** The front end is designed in such a way that it provides an interface which allows the user to interact in an easy manner.

→ **Modularity:** The complete product is broken up into many modules and well-defined interfaces are developed to explore the benefit of flexibility of the product.

→ **Robustness:** This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness.

Non-functional requirements are also called the qualities of a system. These qualities can be divided into execution quality & evolution quality. Execution qualities are security & usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

4.2.2 Organizational Requirements

→ **Process Standards:** IEEE standards are used to develop the application which is the standard used by the most of the standard software developers all over the world.

→ **Design Methods:** Design is one of the important stages in the software engineering process. This stage is the first step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs.

4.2.3 User Requirements

The user requirements document (URD) or user requirements specification is a document usually used to software engineering that specifies the requirements the user expects from software to be constructed in a software project. Once the required information is completely gathered it is documented in a URD, which is meant to spell out exactly what the software must do and becomes part of the contractual agreement. A customer cannot demand feature not in the URD, whilst the developer cannot claim the product is ready if it does not meet an item of the URD.

The URD can be used as a guide to planning cost, timetables, milestones, testing etc. The explicit nature of the URD allows customers to show it to various stakeholders to make sure all necessary features are described. Formulating a URD requires negotiation to determine what is technically and economically feasible. Preparing a URD is one of those skills that lies between a science and economically feasible. Preparing a URD is one of those skills that lies between a science and an art, requiring both software technical skills and interpersonal skills.

4.2.4 Basic Operational Requirements

Operational requirement is the process of linking strategic goals and objectives to tactic goals and objectives. It describes milestones, conditions for success and explains how, or what portion of, a strategic plan will be put into operation during a given operational period, in the case of, a strategic plan will be put into operation during a given operational period, in the case of commercial application, a fiscal year or another given budgetary term. An operational plan is the basis for, and justification of an annual operating budget request. Therefore, a five-year strategic plan would typically require five operational plans funded by five operating budgets.

Operational plans should establish the activities and budgets for each part of the organization for the next 1-3 years. They link the strategic plan with the activities the organization will deliver and the resources required to deliver them. An operational plan draws directly from agency and program strategic plans to describe agency and program missions and goals, program objectives, and program activities. Like a strategic plan, an operational plan addresses four questions:

- Where are we now?
- Where do we want to be?
- How do we get there?

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points:

- **Mission profile or scenario:** It describes about the procedures used to accomplish mission objective. It also finds out the effectiveness or efficiency of the system.
- **Performance and related parameters:** It points out the critical system parameters to accomplish the mission.
- **Utilization environments:** It gives a brief outline of system usage. Finds out

appropriate environments for effective system operation. Operational life cycle: It defines the system lifetime.

4.3 Hardware Requirements

→ Laptop

4.4 Software Requirements

→ Operating system : Windows 10.

→ Coding language : JAVA.

→ Tools : Eclipse Jee Oxygen.

→ Server: Apache Tomcat.

→ Database: MySQL.

Summary

This chapter gives details of the functional requirements, non-functional requirements, resource requirements, hardware requirements, software requirements etc. Again the non-functional requirements in turn contain product requirements, organizational requirements, user requirements, basic operational requirements etc.

Chapter 5

SYSTEM ANALYSIS

5.1 Overview

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

5.2 Feasibility Study

All systems are feasible when provided with unlimited resource and infinite time. But unfortunately, this condition does not prevail in practical world. So it is both necessary and prudent to evaluate the feasibility of the system at the earliest possible time. Months or years of effort, thousands of rupees and untold professional embarrassment can be averted if an illconceived system is recognized early in the definition phase. Feasibility & risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. In this case there are three primary areas of interest:-

5.2.1 Performance Analysis

For the complete functionality of the project work, the project is run with the help of healthy networking environment. Normally, the OS is windows 7. The main theme of this project is to design a system that correctly identifies plant diseases and generates output specific data. Performance analysis is done to find out whether the

proposed system is time efficient and accurate. It is essential that the process of performance analysis and definition must be conducted in parallel.

5.2.2 Technical Analysis

System is only beneficial only if it can be turned into information systems that will meet the organizations technical requirement. Simply stated this test of feasibility asks whether the system will work or not when developed & installed, whether there are any major barriers to implementation. Regarding all these issues in technical analysis there are several points to focus on:-

Changes to bring in the system: All changes should be in positive direction, there will be increased level of efficiency and better customer service.

Required skills: Platforms & tools used in this project are widely used. So the skilled manpower is readily available in the industry.

Acceptability: The structure of the system is kept feasible enough so that there should not be any problem from the users point of view.

5.2.3 Economical Analysis

Economical analysis is performed to evaluate the development cost weighed against the ultimate income or benefits derived from the developed system. For running this system, we simply need a computer. All the features in this system run even on the other Operating Systems. So the system is economically feasible enough.

Summary

The main aim of this chapter is to find out whether the system is feasible enough or not. For these reasons different kinds of analysis, such as performance analysis, technical analysis, economical analysis etc is performed.

Chapter 6

SYSTEM DESIGN

Overview

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customers requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

6.1 System development methodology

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is waterfall model.

6.1.1 Model phases

The waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of

Requirement initiation, Analysis, Design, Implementation, Testing and maintenance.

Requirement Analysis: This phase is concerned about collection of requirement of the system. This process involves generating document and requirement review.

System Design: Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on:- algorithm, data structure, software architecture etc.

Coding: In this phase programmer starts his coding in order to give a full sketch of product. In other words system specifications are only converted in to machine readable compute code. **Implementation:** The implementation phase involves the actual coding or programming of the software. The output of this phase is typically the library, executables, user manuals and additional software documentation .

Testing: In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation.

Maintenance: The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer need, adapt to accommodate change in the external environment, correct errors and oversights previously undetected in the testing phase, enhance the efficiency of the software.

The project abides by prototyping model. The prototyping model is a systems development method in which a prototype is built, tested and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed.

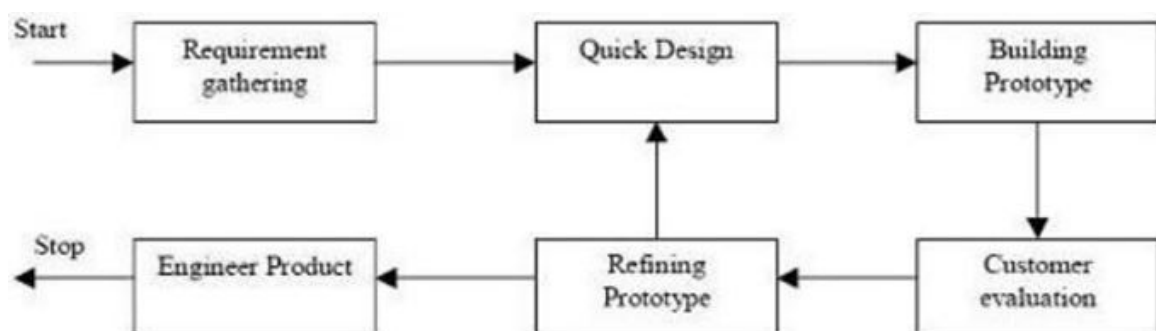


Figure 6.1: Prototyping model

Designing UML diagram specifies, how the process within the system communicates along with how the objects within the process collaborate using both static as well as dynamic UML diagrams since in this ever-changing world of Object Oriented application development, it has been getting harder and harder to develop and manage high quality applications in reasonable amount of time. As a result of this challenge and the need for a universal object modeling language every one could use, the Unified Modeling Language (UML) is the Information industries version of blue print. It is a method for describing the systems architecture in detail. Easier to build or maintains system, and to ensure that the system will hold up to the requirement changes.

6.1.2 Architectural Design

Loyalty engine is designed as an independent system which will offer loyalty solutions to the Card Issuers and Merchants. Loyalty Engine mainly consists of below entities

- Program Manager
- Card Issuers
- Merchants
- Customers



Figure 6.2: Architecture of the entities

Loyalty engine exposes a set of APIs to the card issuers, to the merchants and to the program managers to perform different functionalities required as part of the

loyalty engine.

6.1.2.1 Program Manager

A program manager defines the loyalty programs and runs it in the system. Program manager is responsible for below activities

- Issuer On boarding
- Merchant On boarding
- Loyalty Program Creation and Maintenance
- Settlement between different entities
- Campaign creation for offering discounts, loyalty awards
- Loyalty redemption

The platform supports multiple Program managers who can create multiple loyalty programs.

6.1.2.2 Card Issuer

Using the Loyalty engine, Card Issuers can offer loyalty solutions to their cardholders and offer them with different loyalty rewards. Reward could be in the form of either cash back amounts or it could be in the form of loyalty points which can be accumulated and redeemed for buying variety of items or customer can convert the loyalty points to amount (based on the conversion ratio given in the loyalty program). Each loyalty program is mapped with a list of items or catalogue.

Whenever customer make transactions using their cards, card issuers will award either cash back offers or reward points.

Card Issuer themselves can be program manager and directly manage their own solution or they can work with a program manager to offer loyalty solutions.

6.1.2.3 Merchants

Merchant can also offer loyalty solutions to their frequently visiting customers. Whenever customer makes a purchase at a merchant location, merchant can offer instant discount or loyalty points. In case of merchant based loyalty solutions, there are two types of programs

- Single merchant loyalty program
- Multi-Merchant Loyalty program

In single merchant loyalty program, point accumulation or redemption will happen at the same merchant (in the same store or in any store belonging to the merchant). A customer cannot use the points awarded at a merchant place in some other merchant place.

In case of multi merchant loyalty program, a group of merchants come together and run the loyalty program. Loyalty points issued by one merchant will be valid for use at all the participating merchants.

6.1.2.4 Customers

Customers are the cardholders or account holders in case of Issuer Loyalty Programs and in case of merchant loyalty programs they are the customers of the merchant who makes purchases at the merchant location. A customer is tied to a loyalty program and belongs to the program manager.

Summary

This chapter mainly concentrates on few fundamental design concepts such as system development methodology, system architecture.

Chapter 7

IMPLEMENTATION

7.1 Loyalty-Issuer Portal

The Loyalty-Issuer portal is used to create a loyalty program of what we are going to use along with the Reward catalog where the menu of the catalog items is being displayed which the customers can select these items and redeem their loyalty points. The main entities of the Loyalty-Issuer portal are Program Manager, Issuer, Message, Campaign, Reward Catalog and the Loyalty program. The flow diagram of the issuer loyalty portal is as shown below.

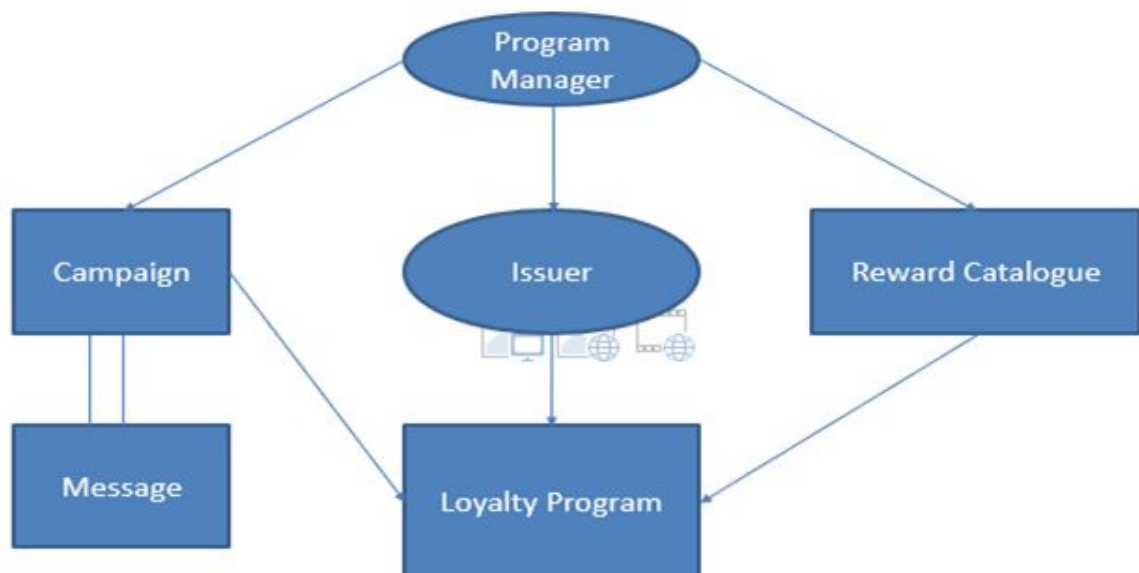


Figure 7.1: Loyalty-Issuer Portal Flow Diagram

The Login page of the Loyalty-Issuer portal is as given below. The login page contains a user-name and password. The user name is unique it may be a valid Email ID or the name of the person with some special characters. When we login to the

portal we will get the dashboard which have the options to create the entities of Program Manager , Issuer , Message , Campaign , Reward Catalog and the Loyalty Program.

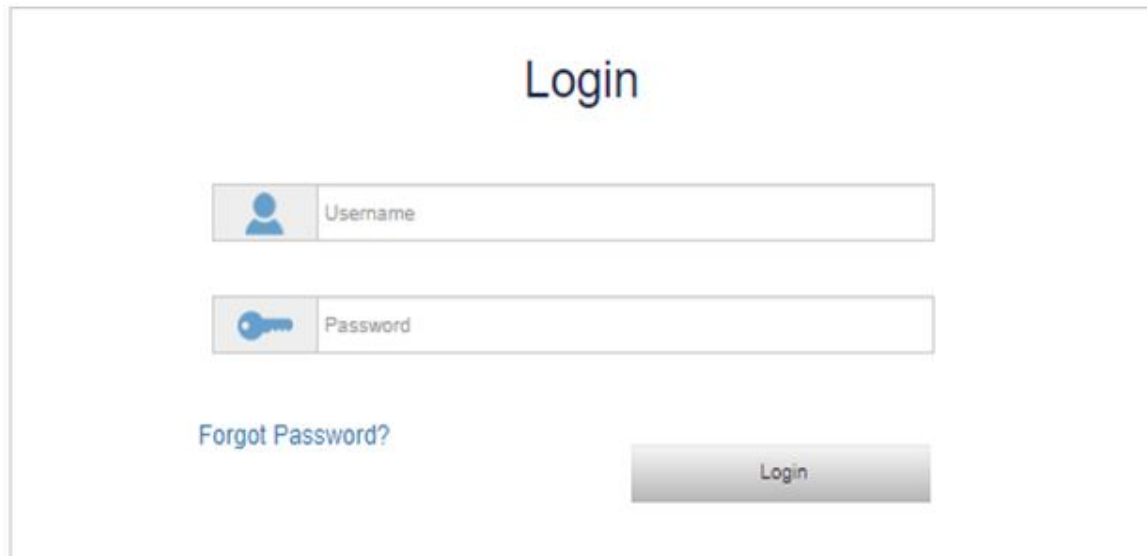


Figure 7.2: Login page



Figure 7.3: Dash Board

There are six important steps to create a loyalty program in the loyalty issuer portal and they are as given below.

- 1.Create a program manager.
- 2.Create an issuer.
- 3.Create a message.
- 4.Create a campaign.
- 5.Create a reward catalog.
- 6.Create a Loyalty program.

7.1.1 Step-1: Create a program manager

A program manager defines the loyalty program and maintains the loyalty program. A program manager can have many number of card issuers and merchants under him and each card issuer may have a different loyalty program compared to other customers. The Program manager is responsible for the below activities.

→ **Issuer on boarding:** He is responsible for all the card issuers who are connected with him.

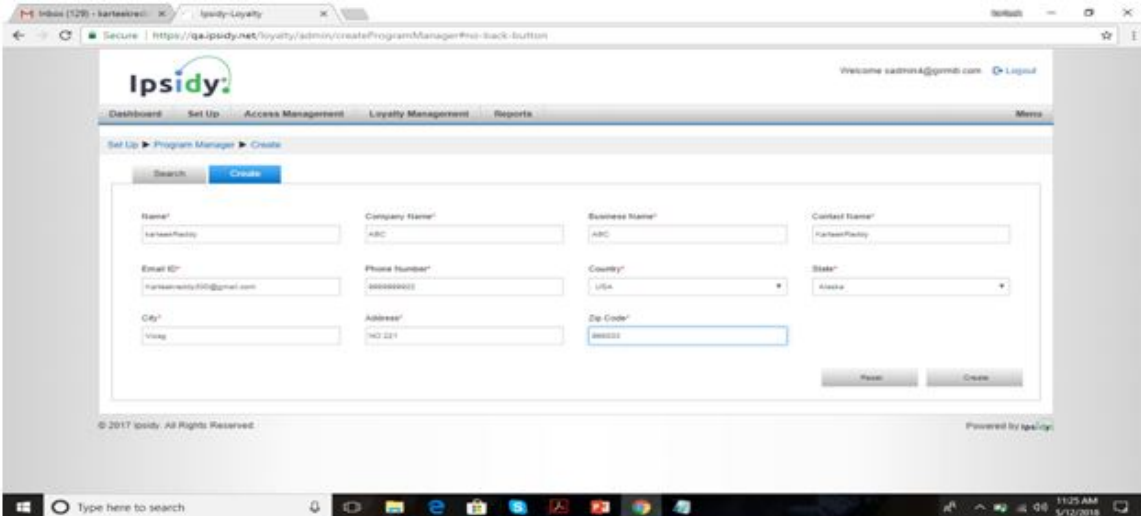
→ **Merchants on boarding:** He is responsible for all the merchants who are connected with him.

→ **Loyalty Program Creation and Maintenance:** He is responsible for creation and maintenance of the different loyalty programs for each card issuer and merchants.

→ **Campaign creation:** He is responsible for the creation of campaigns and to decide the conversion ratio of the amount and the loyalty points.

→ **Loyalty redemption:** He is responsible for the redemption of the loyalty points to buy items in the catalog or to convert the points in to money.

Creation of the Program Manager is as given below.



The screenshot displays the 'Create Program Manager' form within the Ipsidy Loyalty Management System. The form is titled 'Set Up > Program Manager > Create' and includes a search bar and a 'Create' button. The form fields are organized into a grid:

Name*	Company Name*	Business Name*	Contact Name*
<input type="text" value="KartarPacdy"/>	<input type="text" value="ABC"/>	<input type="text" value="ABC"/>	<input type="text" value="KartarPacdy"/>
Email ID*	Phone Number*	Country*	State*
<input type="text" value="KartarPacdy10@gmail.com"/>	<input type="text" value="9888888888"/>	<input type="text" value="USA"/>	<input type="text" value="Alaska"/>
City*	Address*	Zip Code*	
<input type="text" value="Veng"/>	<input type="text" value="HO 221"/>	<input type="text" value="98888"/>	

At the bottom of the form, there are 'Reset' and 'Create' buttons. The footer of the page includes the copyright notice '© 2017 Ipsidy. All Rights Reserved.' and 'Powered by Ipsidy'.

Figure 7.4: Creation of Program Manager

7.1.2 Step-2: Create a Card-Issuer

Step-2: Create a Card-Issuer Using the Loyalty engine, Card Issuers can offer loyalty solutions to their cardholders and offer them with different loyalty rewards. Reward could be in the form of either cash back amounts or it could be in the form

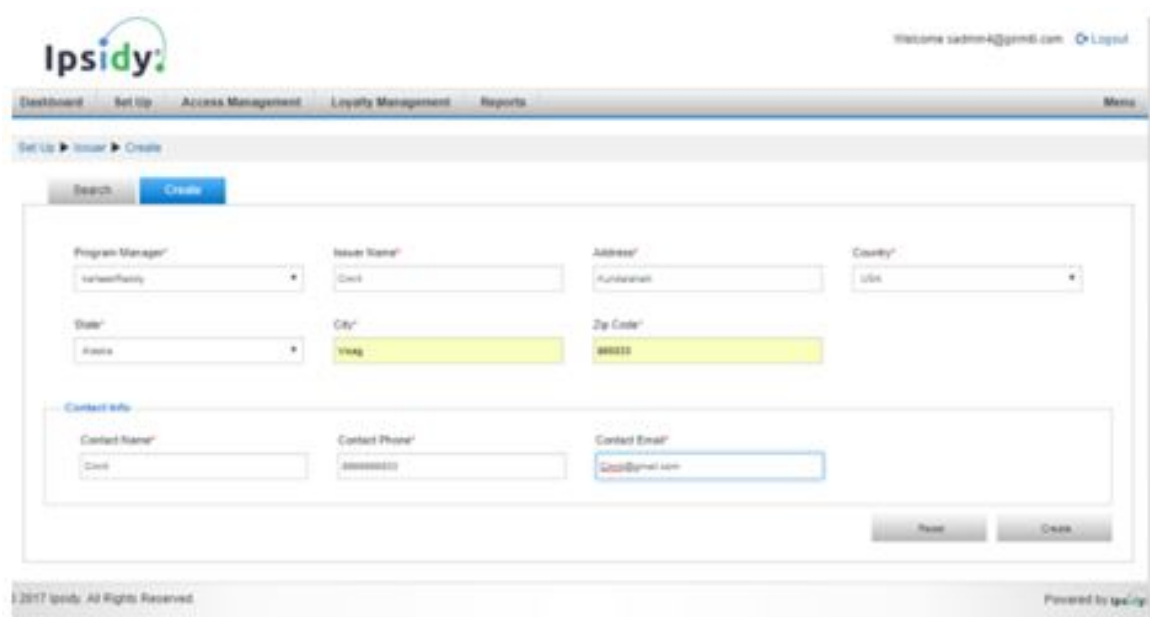
of loyalty points which can be accumulated and redeemed for buying variety of items or customer can convert the loyalty points to amount (based on the conversion ratio given in the loyalty program). Each loyalty program is mapped with a list of items or catalogue. Whenever customer make transactions using their cards, card issuers will award either cash back offers or reward points. Card Issuer themselves can be program manager and directly manage their own solution or they can work with a program manager to offer loyalty solutions. Card Issuer is responsible for the below activities:

→ **BIN to Loyalty Program mapping:** To connect the loyalty program to BIN (Bank Identification Number).

→ **Loyalty Award and redeems:** To offer the loyalty awards and the rewards to the customers.

→ **Campaign to Loyalty Program mapping:** To connect the Loyalty Program to the Campaign so that the conversion ratio is added to the loyalty program.

→ **Customer enrolment:** To enroll new customers.



The screenshot displays the Ipsidy web application interface for creating a Card Issuer. The top navigation bar includes 'Dashboard', 'Set Up', 'Access Management', 'Loyalty Management', 'Reports', and 'Menu'. The main content area shows a 'Set Up' breadcrumb trail leading to 'Issuer' and 'Create'. A search bar and a blue 'Create' button are visible. The form contains the following fields:

- Program Manager: dropdown menu with 'Kunhaman' selected.
- Name: text input with 'Dev'.
- Address: text input with 'Kunhaman'.
- Country: dropdown menu with 'USA'.
- State: dropdown menu with 'Kerala'.
- City: text input with 'Vengal' (highlighted in yellow).
- Zip Code: text input with '686001' (highlighted in yellow).
- Contact Name: text input with 'Dev'.
- Contact Phone: text input with '9896661010'.
- Contact Email: text input with 'dev@gmail.com'.

Buttons for 'Reset' and 'Create' are located at the bottom right of the form. The footer shows '© 2017 Ipsidy. All Rights Reserved.' and 'Powered by ip:ip'.

Figure 7.5: Creation of Card Issuer

7.1.3 Step-3: Create a Message

A Message can be any type of information sent to the customers by the card issuers. The message may consist of the information about the offers, information about the loyalty points, it may be a conformation message etc. the message sent to

the customer may be a simple SMS or an EMAIL or both. The message information will be sent before and during the campaign.

The screenshot shows a web interface for creating a message. At the top, there is a breadcrumb trail: 'Loyalty Management > Message > Create'. Below this, there are two buttons: 'Search' and 'Create'. The main form area contains several sections:

- Message Name:** A text input field containing 'Weekend Sale'.
- Transaction Type:** A dropdown menu showing 'Loyalty Award'.
- Message Type:** Three radio buttons labeled 'SMS', 'EMAIL', and 'Both'. The 'Both' option is selected.
- SMS Content:** A text area containing the text 'Shop for 5000 get 400 loyalty points extra'.
- Email Content:** A text area containing the text 'Shop for 5000 get 400 loyalty points extra'.
- Campaign Info:** A section with a text input field for 'Send Message Before Campaign Start(Days)', two checkboxes for 'Campaign Hold' and 'Campaign Closed', and two buttons at the bottom right: 'Reset' and 'Create'.

Figure 7.6: Creation of a Message

7.1.4 Step-4: Create a Campaign

The main action done in the campaign is creating the conversion ratio of the amount. The conversion ratio is nothing but the number of loyalty points awarded for some unit of amount. Also, the details regarding the date, days and time of the campaign is given. There should be a minimum purchase amount to award the loyalty points in the customers account. That account will also be given while creating the campaign.

7.1.5 Step-5: Create a Reward Catalog

Step-5: Create a Reward Catalog Reward Catalog is nothing but the menu of the Items which we wanted to buy to redeem the loyalty points. The items can be of any thing like some gift vouchers or some items like bags, water bottles etc. The ProgramManager is responsible for the creation on reward catalog and for the menu items.

The Program Manager can also give the pictures of the menu items in the reward catalog, so that the items of the reward catalog are being displayed as the figures to

Figure 7.7: Creation of a Campaign

Figure 7.8: Creating a Reward Catalog

the customers.

7.1.6 Step-5: Create a Loyalty Program

The Loyalty Program is created on the name of Issuer by the Program Manager. There may be different Issuers under different Program Managers so, each issuer has a separate distinct Loyalty Program. The currency in the loyalty Program should be equal to the currency of the issuer that is the important thing to note while creating

a loyalty program. There is an option called Transactions Supported there we have to give all the available options enable the campaign name and catalog name is also included because campaign is used for the conversion ratio of the money and the loyalty points and then the catalog name is used to redeem the loyalty points of the customer by selecting the items present in the menu.

The screenshot shows a web form titled "Loyalty Management > Loyalty > Create". At the top, there are "Search" and "Create" buttons. The form contains several input fields and dropdown menus:

- Loyalty Program Name***: Text input with "Weekend sale".
- Program Currency***: Dropdown menu with "US Dollar".
- Loyalty Program Type***: Dropdown menu with "Issuer Loyalty Program".
- Issue Name***: Dropdown menu with "Cmrt".
- Loyalty Reward Type***: Radio buttons for "Cash Back offer" (selected) and "Loyalty Points Award".
- Transactions Supported***: A multi-select dropdown menu with options: "Loyalty Award", "Loyalty Redeem", "Redeem with top up Cash", "Points conversion", "Balance inquiry", and "Transaction history".
- Campaign Name***: Dropdown menu with "Weekend sale".
- Catalogue Name***: Dropdown menu with "Weekend sale".
- Redeem Points to Amount Conversion Ratio**: A section with two input fields: "Redeem Points*" (containing "20") and "Redeem Amount*" (containing "50").

At the bottom right of the form, there are "Reset" and "Create" buttons.

Figure 7.9: Creation of Loyalty Program

7.2 Loyalty-Admin Portal

The Loyalty-Admin Portal is used to connect the loyalty program from the Loyalty-Issuer portal and to create the customers. The main entity of this portal is a Program manager who is an issuer who creates the customers. There will be a bank, a BIN, program manager, a partner, a sales person who is also known as a merchant, we create a customer. The flow diagram of the issuer loyalty portal is as shown below.

The Login page of the Loyalty-Admin portal is as given below. The login page contains a user-name and password. The user name is unique it may be a valid Email ID or the name of the person with some special characters. When we login to the

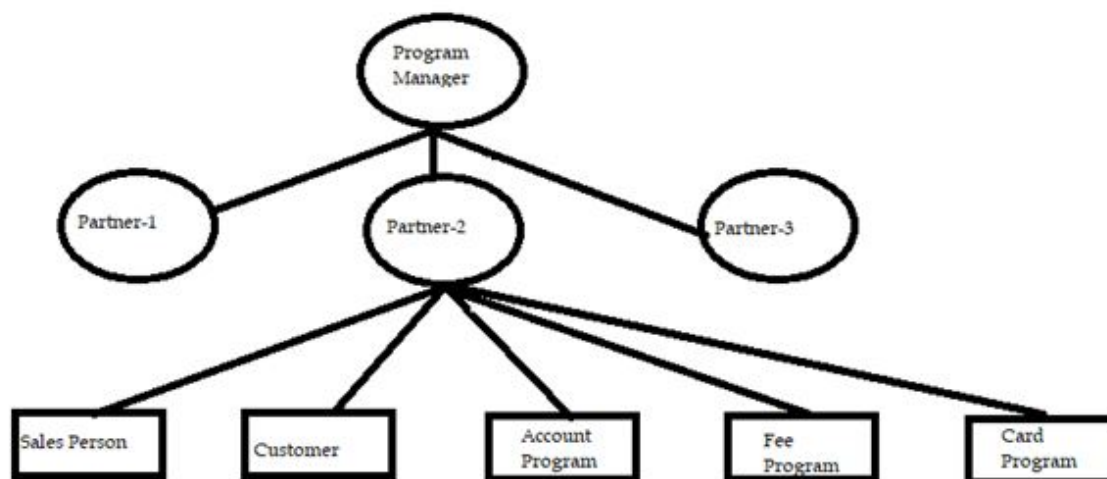


Figure 7.10: Loyalty-Admin Flow Diagram

portal we will get the dashboard which have the options to create the entities of bank, a BIN , program manager , a partner ,a sales person who is also known as a merchant and also a customer. We also create an Account Program, a Card Program and a Fee Program as they are different for different Issuers.

The image shows a login page with the title "Login". It features two input fields: "Username" with a user icon and "Password" with a key icon. Below the password field is a link for "Forgot Password?". A "Login" button is positioned at the bottom right of the form.

Figure 7.11: Login Page of Loyalty-Admin Portal

When we login in to the portal we get a dashboard with all the options to create the entities. The Dashboard of the Loyalty-Admin Portal is as given below.

There are six important steps to create a loyalty program in the loyalty issuer portal and they are as given below.

- 1.Create a Bank.
- 2.Create a BIN.
- 3.Create a Program Manager.
- 4.Create a Partner.
- 5.Create a Sales Person.

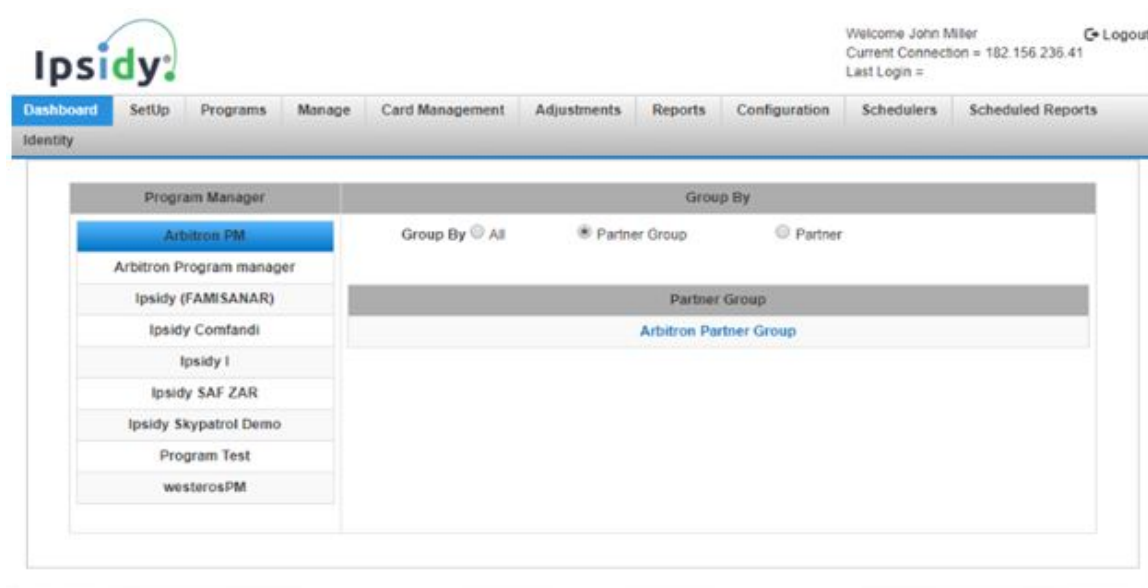


Figure 7.12: Dashboard of Loyalty-Admin Portal

6. Configure the Loyalty Program.
7. Create an Account Program.
8. Create a fee Program.
9. Create a Card Program.
10. Create an Account Program.

7.2.1 Step-1: Create a Bank

The Bank is created initially to convert the loyalty points into the money and to store them in the customer's accounts. Using Banks, the accounts have been created by the partner for the customers and the loyalty points are stored in that account. One customer can have multiple number of accounts and each account has a separate card which is called the loyalty card. The creation of the bank account is as shown in the figure below. The currency should be the same as the Issuer in the Loyalty-Admin portal.

7.2.2 Step-2: Create a BIN

Bin is nothing but the Bank Identification Number. The first six numbers of an Account number are called as the Bank Identification Number. BIN will be the same for all the accounts of the bank. The next three numbers after the BIN is called the BIN extension. BIN is actually A payment card number, primary account number (PAN),
 Dept Of CSE - Feb - May 2018

Set Up ▶ Bank ▶ Edit

Search Edit

Bank Code*	Bank Name*	Settlement Routing Number*	Settlement Account Number*
111222333	Station Bank	00088222	00000888
Address 1*	Address 2	Country*	State*
bangalore	bangalore	US	Alaska
City*	Zip*	Primary Contact Name*	Contact Mobile Number
bangalore	457870	00000000	
Contact Phone Number*	Extension	Fax	E-mail Address*
000-888-5555	Numerics only	Numerics only	kartikreddy30@gmail.com
Status*			
Active			

Cancel Update

Figure 7.13: Creation of a bank

or simply a card number, is the card identifier found on payment-cards, such as credit cards and debit cards, as well as stored value cards, gift cards and other similar cards. In some situations, the card number is referred to as a bank card number. The card number is primarily a card identifier and does not directly identify the bank account number/s to which the card is/are linked by the issuing company. The card number prefix identifies the issuer of the card, and the digits that follow are used by the issuing organization to identify the cardholder as a customer and which is then associated by the issuing organization with the customer's designated bank accounts. Creation of bin is as given in the diagram below.

Set Up ▶ BIN ▶ View

Search View

BIN Type*	BIN*	Bank Name*	BIN Effective Date (DDMM/YYYY)
Prepaid	000000	Station Bank	14/05/2018

Cancel

Figure 7.14: Creation of BIN

7.2.3 Step-3: Create a Program Manager

A program manager defines the loyalty program and maintains the loyalty program. A program manager can have many number of partners and sales persons under him and each partners may have a different loyalty program compared to other customers. The Program manager in this portal means the Issuer who can create the customers and issue them the loyalty cards and loyalty points. The Creation of the Program Manager is as given below.

The screenshot shows a web form titled "Set Up Program Manager Edit". The form contains the following fields and controls:

- Program Manager Name***: Text input with value "Station Progm Manager".
- Company Name***: Text input with value "ABC".
- Business Entity Name***: Text input with value "ABC".
- Contact Person***: Text input with value "KarteesReddyAdmin".
- Contact Number***: Text input with value "888-999-7777".
- Extension**: Text input with value "Numerics only".
- E-mail Address***: Text input with value "karteesreddy30@gmail.com".
- Bank Name***: Dropdown menu with options: Citibank Colombia, ITAU Colombia, South Africa Bank, Station Bank.
- Currency***: Dropdown menu with value "US Dollar".
- Program Manager Logo**: Text input with value "Browse" and a file upload icon.
- Default Program Manager**: Checkable field.
- To View Program Manager Logo**: Link with text "Click Here".
- Buttons**: "Cancel" and "Update" buttons at the bottom right.

Figure 7.15: Creation of the Program Manager

7.2.4 Step-4: Create a Partner

The Partner is an assistant to the Programmer who has a particular customer group under him. There will be a finite set of partners under a particular Program Manager. Each partner is belonging to one company which is different from each other. A Partner will have a different Account, Card and Fee program compared to other partners. Creation of a Partner is as given Below.

7.2.5 Step-5: Create a Sales Person

A Sales Person is also called as a merchant. Merchant can also offer loyalty solutions to their frequently visiting customers. Whenever customer makes a purchase at a merchant location, merchant can offer instant discount or loyalty points. In case

The screenshot shows a web interface for creating a partner. At the top, there are navigation links: 'Set Up', 'Partner', and 'Edit'. Below these are 'Search' and 'Edit' buttons. The main form is organized into several sections:

- Program Manager Name***: Station Progm Manager
- Bank Name**: Station Bank
- Currency***: US Dollar
- Client ID***: 123456789
- Company Name***: ABC
- Business Entity Name***: ABC
- Partner Name***: KartekReddyPartner
- Contact Person***: Station
- Contact Number***: 777-888-9999
- Extension**: Numerics only
- Address 1***: Bangalore
- Address 2**: bangalore
- Country**: US
- State**: Alaska
- City***: bangalore
- Zip***: 457679
- Partner Logo**: Browse...
- To View Partner Logo Click Here**: [To View Partner Logo Click Here](#)
- Enable API Service**: Yes
- IDC Partner**: Yes

Figure 7.16: Creation of a Partner

of merchant-based loyalty solutions, there are two types of programs

→ Single merchant loyalty program → Multi-Merchant Loyalty program In single merchant loyalty program, point accumulation or redemption will happen at the same merchant (in the same store or in any store belonging to the merchant). A customer cannot use the points awarded at a merchant place in some other merchant place. In case of multi merchant loyalty program, a group of merchants come together and run the loyalty program. Loyalty points issued by one merchant will be valid for use at all the participating merchants. Creation of Murchant is as shown below.

7.2.6 Step-6: Configure the Loyalty Program

The Loyalty Program configuration is the most important step. We have to configure the loyalty program of Loyalty-Issuer portal to the Loyalty-Admin portal. The Loyalty ID will be generated in the Loyalty-Issuer portal when we are creating the Issuer. That Loyalty ID is given as the input to configure the Loyalty Program and we also give the URL of the Loyalty-Issuer portal. Configuration is as given below.

Set Up > Sales Person > Edit

Search Edit

Basic Information

Sales Person Name*

Stalin Sales Person

Select Program Manager

Program Manager Name*

Arbizon PM

Arbizon Program manager

Issudy (FAM/SAHAR)

Issudy Comland

This field is mandatory

Address and Contact Details

Address 1* Address 2* Country* State*

bangalore bangalore US Alaska

City* Zip* Contact Phone Number* Extension

bangalore 457879 900-888-8888 Numerics only

Contact Mobile Number* E-mail Address* Edit TIN/SSN Yes TIN/SSN*

900-777-8888 kartasekady30@gmail.com 00004321

Fax

Numerics only

Bank Details

Bank Name* Country* State* City*

Stalin Bank US Alaska bangalore

Routing Number* Account Number* Bank Account Type* Status*

909993222 999999999 Savings Account Active

Cancel Update

Figure 7.17: Creation of a Sales Person

7.2.7 Step-7: Create the Account Program

In the Loyalty-Admin portal there are a finite number of partners where each partner will have a different functionality. The functionality of accounts used by all the partners should be different from one other. So, Program Manager creates the account program by giving the partner name and the account details. The card load limit is the number of loyalty points the partner can issue to his/her customer at the maximum and that should be unlimited by default.

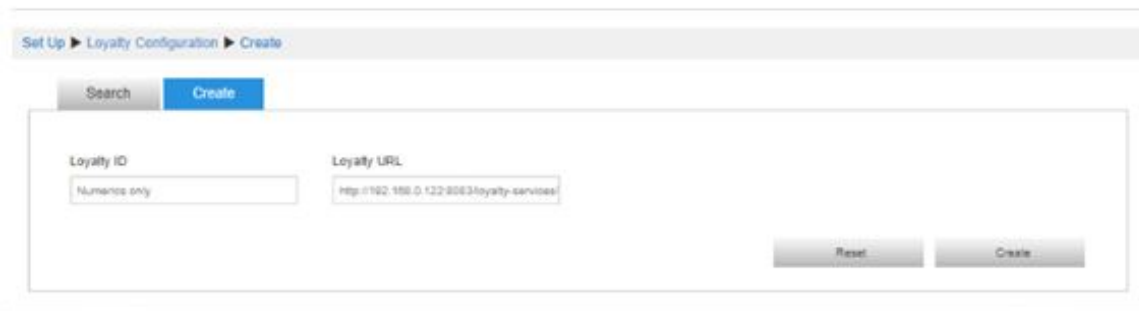


Figure 7.18: Configuration of the loyalty program

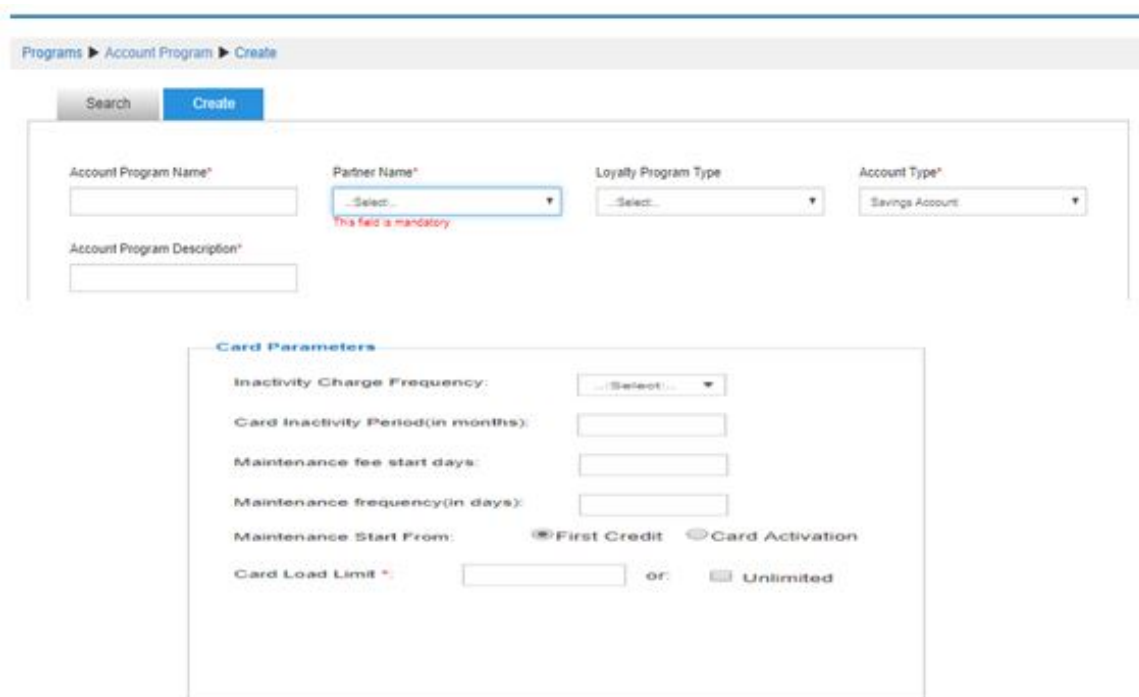


Figure 7.19: Creation of the Account Program

7.2.8 Step-8: Create the Fee Program

In the Loyalty-Admin portal there are a finite number of partners where each partner will have a different functionality. The functionality of Fee used by all the partners should be different from one other. Fee is used to redeem some amount of money when the customer redeems some points, when the customer top-up the cash etc. So, Program Manager creates the Fee program by giving the partner name and the account details. We have to add all the needed fee options of the fee depending on the functionality of the partner.

The screenshot shows a web interface for creating a fee program. At the top, there are 'Search' and 'Create' buttons. Below, the form is organized into several sections:

- Fee Program Name***: A text input field.
- Fee Program Description***: A text input field.
- Partner Name***: A dropdown menu with a 'Select...' option.
- Account Program Name***: A dropdown menu with a 'Select...' option.
- Account Currency**: A dropdown menu with a 'Select...' option.
- PM % Share***: A text input field.
- Partner % share***: A text input field.
- Transaction Types**: A list of transaction types on the left, with 'POS_PURCHASE - INVALID_CARD' highlighted. A right arrow button is next to the list, and a left arrow button is below it.

Figure 7.20: Creation of the Fee Program

7.2.9 Step-9: Create the Card Program

In the Loyalty-Admin portal there are a finite number of partners where each partner will have a different functionality. The functionality of cards used by all the partners should be different from one other. The card might be a food card or a shopping card etc. depending on the partners functionality. So, Program Manager creates the card program by giving the partner name and the account details. The IIN (Issuer Identification number) is the important field where we give the ID of the issuer .

7.2.10 Step-10: Create the Customer

Customers are the cardholders or account holders in case of Issuer Loyalty Programs and in case of merchant loyalty programs they are the customers of the merchant who makes purchases at the merchant location. A customer is tied to a loyalty program and belongs to the program manager. Whenever an Issuer is on boarded, issuer will be giving requests to the program manager to enroll the customer in to the loyalty program that are mapped to the issuer. When a customer is enrolled a loyalty account is created for the customer using the Loyalty Program Name provided by the Issuer. There are several fields for the customer and they are shown below.

The type of the customer is always the ipsidy customer. Now, the program manager map all the entities that are necessary to create a customer and the currency

Figure 7.21: Creation of Card Program

here is the most important part. The currency should be same as that of the card issuer.

The address of the customer is given and it is the mandatory field where all the details of the customer is given.

The customers Identity number should be linked to verify that the customer details which are given are valid.

On clicking on create the mail will be sent to the valid customer with his details of his username and a temporary password and a link will be present which says click here. The customer can click on the link and the customer portal will be opened.

The screenshot shows a 'Create' form for a customer. At the top, there are 'Search' and 'Create' buttons. Below is a 'Customer Type' section, followed by 'User Information'. The form is divided into three main sections: Personal Information, Country, and Contact.

Personal Information:

- Program Manager Name*: Ramsay PM (dropdown)
- Partner Name*: Ramsay Partner (dropdown)
- Customer Group*: Ramsay Customer Group (dropdown)
- Title: Mr (dropdown)
- First Name*: Kartek (text input)
- Middle Name/Initial: (text input)
- Last Name*: Reddy (text input)
- Company Name: (text input)
- Date of Birth*(DD/MM/YYYY): 19/05/1997 (calendar icon)
- Login User Name*: KartekReddy (text input)

Country:

- Country of Citizenship: South Africa (dropdown)
- Country of Residence*: South Africa (dropdown)
- SSN/National ID Number*: 5544 (text input)

Contact:

- Email Address*: kartekreddy30@gmail.com (text input)
- Confirm E-mail Address*: kartekreddy30@gmail.com (text input)
- Mobile Number*: 995555552 (text input)
- Extension: (text input with 'Numerics only' placeholder)

Figure 7.22: Mapping the Entities

The screenshot shows the 'Details of the customer' form. It includes sections for 'Physical Address' and 'Mailing Address'. The 'Mailing Address' section has a checkbox 'Same as Physical Address' which is checked.

Physical Address:

- Address Line 1*: Bangalore (text input)
- Address Line 2: (text input)
- City*: Marshall (text input)
- Zip/Postal Code*: 624438 (text input)
- Country*: South Africa (dropdown)
- State/Province*: Northern Cape (dropdown)

Mailing Address:

- Address Line 1*: Bangalore (text input)
- Address Line 2: (text input)
- City*: Marshall (text input)
- Zip/Postal Code*: 624438 (text input)
- Country*: South Africa (dropdown)
- State/Province*: Northern Cape (dropdown)

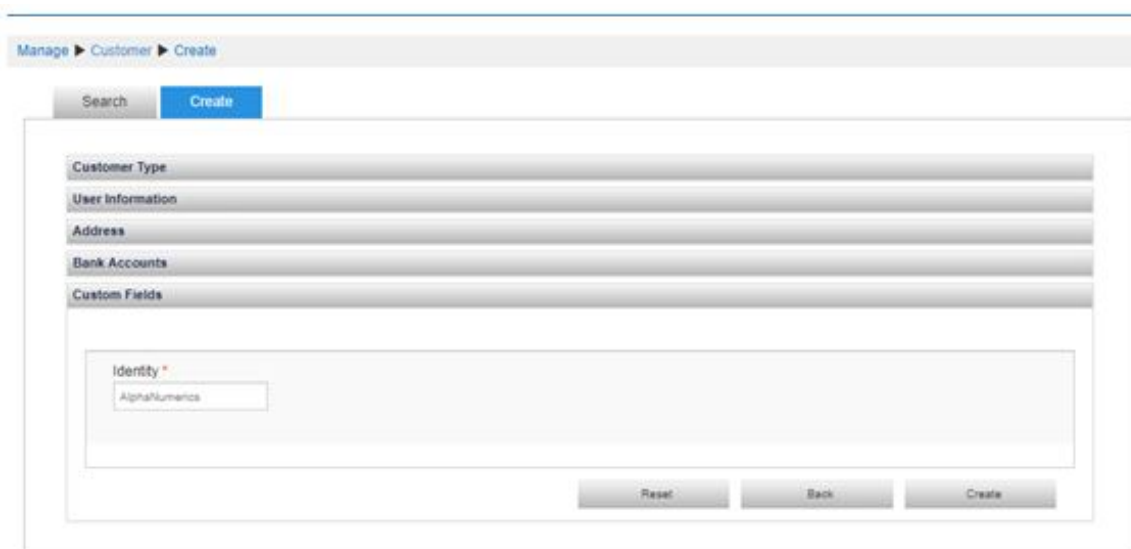
At the bottom, there are 'Reset', 'Back', and 'Next' buttons. Below the address sections are 'Bank Accounts' and 'Custom Fields' sections.

Figure 7.23: Details of the customer

7.3 Loyalty-Customer Portal

The loyalty customer portal is for the customers to access their account and redeem the points to buy the products in the catalog or to convert those points into money. The login page of the portal is as given below.

When the customer logs in into the portal the details of his account and the details of his loyalty points will be displayed on the dashboard along with the last transaction history.



The screenshot shows a web interface for creating a customer. At the top, there is a breadcrumb trail: "Manage > Customer > Create". Below this, there are two tabs: "Search" and "Create", with "Create" being the active tab. The main content area is a form with several sections: "Customer Type", "User Information", "Address", "Bank Accounts", and "Custom Fields". The "Custom Fields" section is expanded, showing a field labeled "Identity *" with a dropdown menu set to "AlphaNumerics". At the bottom of the form, there are three buttons: "Reset", "Back", and "Create".

Figure 7.24: Identity of the customer

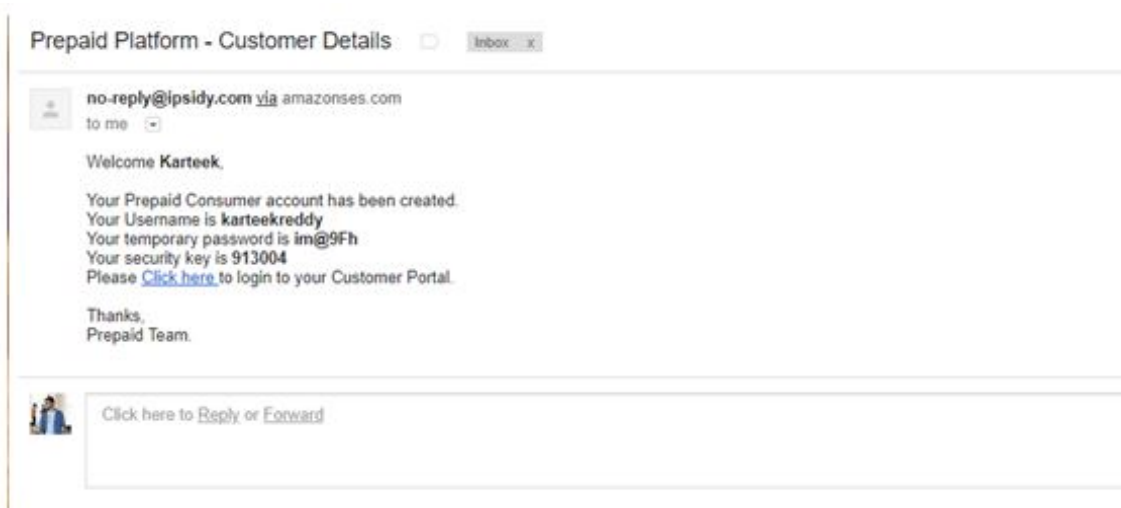


Figure 7.25: Email generated to the customer

To view the no of loyalty points the customer has, the customer should click on the click here to view points and then the loyalty points the customer have will be displayed.

When we click on the redeem option we will go to the next page and there we have two options redeem for full points and redeem points for top-up. When we click on the first option, the catalogue which we have created will be displayed. The customer will click on the items he wanted to and he can redeem the loyalty points.

Figure 7.26: Loyalty-Customer portal Login page

Account Summary						
Customer Account	Account Name	Available Balance (ZAR)	Current Balance (ZAR)	Status	Card Number	Total Loyalty Points
2000011944	NA	999,100.00	999,100.00	Active	7000-XXXX-XXXX-8237	Click here to view points

News Updates		Total Count :
News Title	News Summary	
No Record found		

Recent Transactions								
System Transaction Time	Device Local Transaction Time	Account Number	Description	Transaction Type	Amount (ZAR)	Available Balance (ZAR)	Current Balance (ZAR)	Status
15/05/2018 09:43:19		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	30.00	999,100.00	999,100.00	Executed
15/05/2018 08:18:46		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	10.00	999,070.00	999,070.00	Executed
14/05/2018 06:28:52		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	30.00	999,060.00	999,060.00	Executed
14/05/2018 05:26:10		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	30.00	999,030.00	999,030.00	Executed

Figure 7.27: Dashboard of the Loyalty-Customer portal

When we select on the redeem points to top-up, the customer will be asked to enter the no of points to be converted in to currency. When the customer enters the amount of loyalty points to be converted in to money and they will be transferred in to the customers bank account which is linked.

After the transaction is done, the customer can view his history to conform weather the loyalty points have been converted or not.

Account Summary						
Customer Account	Account Name	Available Balance (ZAR)	Current Balance (ZAR)	Status	Card Number	Total Loyalty Points
2000011944	NA	999,100.00	999,100.00	Active	7000-XXXX-XXXX-8237	300 Redeem


Figure 7.28: Loyalty points Display

My Profile ▶ My Profile


Loyalty Points

Customer Account Number


Loyalty Total Points



ItemPoints 1
ItemCode 0001
ItemDescription Coffee



ItemPoints 2
ItemCode 0002
ItemDescription Soft Drink



ItemPoints 3
ItemCode 0003
ItemDescription Sweet

Figure 7.29: Catalog to redeem the loyalty points

My Profile ▶ My Profile

Loyalty Points

Customer Account Number: 2000011944 Loyalty Total Points: 300

Redeem For Full Points
 Redeem Points Convert To TopUp

Enter Point To Redeem: 20

Cancel Submit

Figure 7.30: Conversion of loyalty points in to money

Recent Transactions								
System Transaction Time	Device Local Transaction Time	Account Number	Description	Transaction Type	Amount (ZAR)	Available Balance (ZAR)	Current Balance (ZAR)	Status
22/05/2018 02:19:31		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	20.00	999,120.00	999,120.00	Executed
15/05/2018 09:43:19		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	30.00	999,100.00	999,100.00	Executed
15/05/2018 08:18:46		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	10.00	999,070.00	999,070.00	Executed
14/05/2018 06:28:52		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	30.00	999,060.00	999,060.00	Executed
14/05/2018 05:26:10		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	30.00	999,030.00	999,030.00	Executed

Figure 7.31: Transaction history

Chapter 8

TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve:-

- To affirm the quality of the project.
- To find and eliminate any residual errors from previous stages.
- To validate the software as a solution to the original problem.
- To provide operational reliability of the system.

8.1 Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are:

8.1.1 White box testing

White box testing (clear box testing, glass box testing, and transparent box testing or structural testing) uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. While white box testing is applicable at the unit, integration and system levels of the software testing process, it is typically applied to the unit. While it normally tests paths within a unit, it can also

test paths between units during integration, and between subsystems during a system level test

Though this method of test design can uncover an overwhelming number of test cases, it might not detect unimplemented parts of the specification or missing requirements, but one can be sure that all paths through the test object are executed. Using white box testing we can derive test cases that:

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structure to assure their validity .

8.1.2 Black box testing

Black box testing focuses on the functional requirements of the software. It is also known as functional testing. It is a software testing technique whereby the internal workings of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs.

The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specifications. It enables us to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing is an alternative to white box technique. Rather it is a complementary approach that is likely to uncover a different class of errors in the following categories:-

- Incorrect or missing function.
- Interface errors.
- Performance errors.
- Initialization and termination errors.

Advantages

- The test is unbiased as the designer and the tester are independent of each other.
- The tester does not need knowledge of any specific programming languages.
- The test is done from the point of view of the user, not the designer.

8.2 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Test Id	Component	Test Done	Result
1	Login	Test to check whether the login is done or not	Success.
2	Issuer	Test to check whether the Issuer is successfully created or not	Success.
3	Campaign	Test to check whether Campaign is created or not	Success.
4	Reward Catalog	Test to check whether reward items are added to menu or not	Success.
5	Loyalty	Test to check whether the loyalty program is configured or not	Success

Table 8.1: Functions for Input

8.3 Integration Testing

Upon completion of unit testing, integration testing begins. Individual modules are combined and tested as a group. Integration testing is black box testing. The purpose of integration testing is to ensure distinct components of the application still work in accordance to user requirements. Integration testing is considered complete, when actual results and expected results are either in line or differences are explainable based on client input. It concentrates on data transfer between modules. Integration

Test Id	Component	Test Done	Result
1	Customer	Test to check whether the mail has been sent to the customer or not	Success.
2	Account	Test to check whether the loyalty account is linked with bank account or not	Success.
3	Redemption	Test to check whether loyalty points are redeemed or not	Success.
4	Topup	Test to check whether the loyalty points are converted to money or not	Success.

Table 8.2: Functions for Output

testing is a logical extension of unit testing. Two units that have already been tested are combined into a component and the interface between them is tested. Integration testing identifies problems that occur when units are combined. The errors that arise can be attributed to those occurring due to the combination of modules, resulting from errors across interface.

The Integration Testing fig 8.3 shows the functions that are combined into different classes and the module as a whole tested for its functionality. Finally, all the modules are integrated and tested. This is important to check for error-free interaction between various classes and its modules. The integration testing table shows the important modules integrated.

Module	Functions Integrated	Test Done
Issuance Module	Functions used to create the entities and loyalty programs in the module	Tested the function of issuance module.
Admin Module	Functions used to create the entities and configure the loyalty programs in the module	Tested the function of admin module.
Customer Module	Functions used in displaying the reward catalog and transaction history	Tested the function of customer module.

Table 8.3: Integration Testing

8.4 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "interassemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specifications.

The following examples are different types of testing that should be considered during System testing:

- Graphical user interface testing.
- Usability testing .
- Software performance testing.
- Compatibility testing .
- Exception handling .
- Load testing .
- Volume testing.

Although different testing organizations may prescribe different tests as part of System testing, this list serves as a general framework or foundation to begin with.

8.5 Quality Assurance

Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confidence that the product quality is meeting its goals. This is an umbrella activity that is applied throughout the engineering process. Software quality assurance encompasses:-

- Analysis, design, coding and testing methods and tools.
- Formal technical reviews that are applied during each software engineering.
- Multitier testing strategy.
- Control of software documentation and the change made to it.
- A procedure to ensure compliance with software development standards.
- Measurement and reporting mechanisms.

Quality Assurance (QA) is a way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to customers. QA is applied to physical products in pre-production to verify what will be made meets specifications and requirements, and during manufacturing production runs by validating lot samples meet specified quality controls. QA is also applied to software to verify that features and functionality meet business objectives, and that code is relatively bug free prior to shipping or releasing new software products and versions. Quality Assurance refers to administrative and procedural activities implemented in a quality system so that requirements and goals for a product, service or activity will be fulfilled. It is the systematic measurement, comparison with a standard, monitoring of processes and an associated feedback loop that confers error prevention. This can be contrasted with quality control, which is focused on process output.

Two principles included in Quality Assurance are: "Fit for purpose", the product should be suitable for the intended purpose; and "Right first time", mistakes should be eliminated. QA includes management of the quality of raw materials, assemblies, products and components, services related to production, and management, production and inspection processes. Suitable quality is determined by product users, clients or customers, not by society in general. It is not related to cost, and adjectives or descriptors such as "high" and "poor" are not applicable. For example, a low priced product may be viewed as having high quality because it is disposable, where another may be viewed as having poor quality because it is not disposable.

Software quality assurance (SQA) consists of a means of monitoring the software

engineering processes and methods used to ensure quality. The methods by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards, such as ISO 9000 or a model such as CMMI.SQA encompasses the entire software development process, which includes processes such as requirements definition, software design, coding, source code control, code reviews, software configuration management, testing, release management, and product integration.

8.5.1 Quality Factor

An important objective of quality assurance is to track the software quality and assess the impact of methodological and procedural changes on improved software quality. The factors that affect the quality can be categorized into two broad groups:

1. Factors that can be directly measured.
2. Factors that can be indirectly measured.

These factors focus on three important aspects of a software product

- Its operational characteristics.
- Its ability to undergo changes.
- Its adaptability to a new environment.
- Effectiveness or efficiency in performing its mission.
- Duration of its use by its customer.

In the context of software engineering, software quality refers to two related but distinct notions that exist wherever quality is defined in a business context: Software functional quality reflects how well it complies with or conforms to a given design, based on functional requirements or specifications. That attribute can also be described as the fitness for purpose of a piece of software or how it compares to competitors in the marketplace as a worthwhile product; Software structural quality refers to how it meets non-functional requirements that support the delivery of the functional requirements, such as robustness or maintainability, the degree to which the software was produced correctly.

Structural quality is evaluated through the analysis of the software inner structure, its source code, at the unit level, the technology level and the system level, which in effect how its architecture adheres to sound principles of software architecture outlined

in a paper on the topic by OMG. In contrast, functional quality is typically enforced and measured through software testing. Historically, the structure, classification and terminology of attributes and metrics applicable to software quality management have been derived or extracted from the ISO 9126-3 and the subsequent ISO 25000:2005 quality model, also known as SQuaRE. Based on these models, the Consortium for IT Software Quality (CISQ) has defined five major desirable structural characteristics needed for a piece of software to provide business value: Reliability, Efficiency, Security, Maintainability and (adequate) Size.

Software quality measurement quantifies to what extent a software or system rates along each of these five dimensions. An aggregated measure of software quality can be computed through a qualitative or a quantitative scoring scheme or a mix of both and then a weighting system reflecting the priorities. This view of software quality being positioned on a linear continuum is supplemented by the analysis of "critical programming errors" that under specific circumstances can lead to catastrophic outages or performance degradations that make a given system unsuitable for use regardless of rating based on aggregated measurements.

Such programming errors found at the system level represent up to 90 percent of production issues, whilst at the unit-level, even if far more numerous, programming errors account for less than 10 percent of production issues. As a consequence, code quality without the context of the whole system, as W. Edwards Deming described it, has limited value.

To view, explore, analyze, and communicate software quality measurements, concepts and techniques of information visualization provide visual, interactive means useful, in particular, if several software quality measures have to be related to each other or to components of a software or system.

Summary

The chapter discusses the tests that are done on the system to check its functionality. Testing is carried out at three different levels from the module level to the system level checking for errors at each stage. The remarks have also been documented.

Chapter 9

RESULTS & EXECUTION

The snap shots of the results is as shown in the figure below. The Transactions at the customer loyalty portal is as shown below for both Reward catalog and Top-up cash.

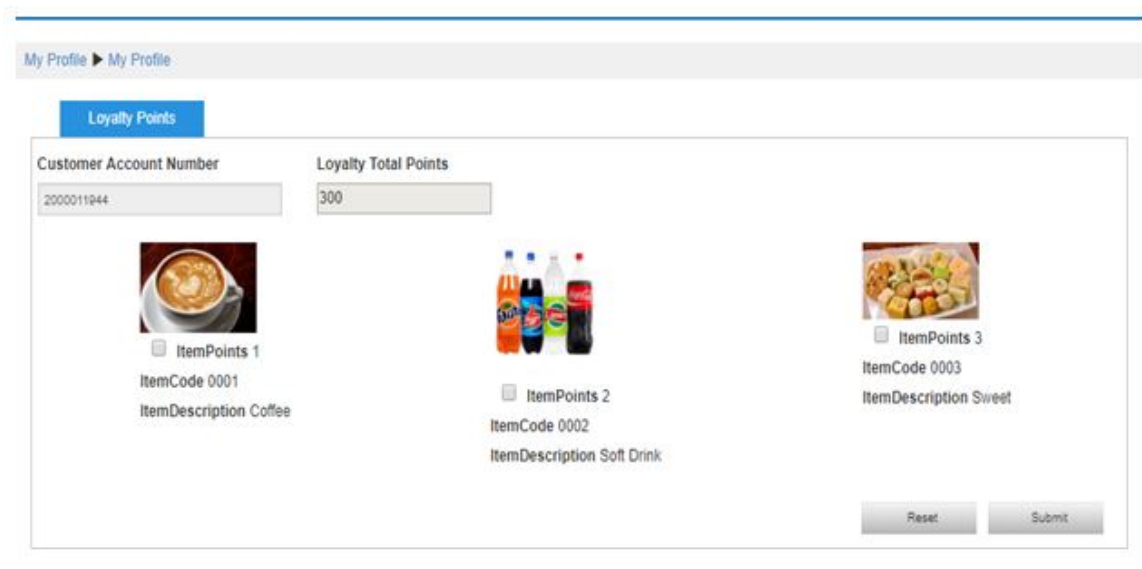


Figure 9.1: Redemption of loyalty points using the Reward Catalog

When we click on the redeem option we will go to the next page and there we have two options redeem for full points and redeem points for top-up. When we click on the first option, the catalogue which we have created will be displayed. The customer will click on the items he wanted to and he can redeem the loyalty points.

When we select on the redeem points to top-up, the customer will be asked to enter the no of points to be converted in to currency. When the customer enters the amount of loyalty points to be converted in to money and they will be transferred in

Figure 9.2: Redemption of loyalty points using the top-up cash

to the customers bank account which is linked.

After the transaction is done, the customer can view his history to conform weather the loyalty points have been converted or not.

Recent Transactions								
System Transaction Time	Device Local Transaction Time	Account Number	Description	Transaction Type	Amount (ZAR)	Available Balance (ZAR)	Current Balance (ZAR)	Status
22/05/2018 02:19:31		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	20.00	999,120.00	999,120.00	Executed
15/05/2018 09:43:19		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	30.00	999,100.00	999,100.00	Executed
15/05/2018 08:18:46		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	10.00	999,070.00	999,070.00	Executed
14/05/2018 06:28:52		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	30.00	999,060.00	999,060.00	Executed
14/05/2018 05:26:10		2000011944	Amount Credited through Redeeming Loyalty Points	Credit	30.00	999,030.00	999,030.00	Executed

Figure 9.3: Transaction history

Chapter 10

CONCLUSION & FUTURE ENHANCEMENT

10.1 Conclusion

Loyalty Engine offers Loyalty solutions to card issuers and Merchants. loyalty engine should be a multitenant solution and should support multiple loyalty programs defined by different card issuers and different merchants. The platform should support creation and management of multiple loyalty programs seamlessly.

10.2 Future Enhancement

In future, the loyalty points can be transferred from one customer to the other so that the customers who will not redeem the points can transfer the loyalty points to the customers who will make use of the loyalty points.

REFERENCES

- [1] Su Lujun. Study on the Mechanism of Service Fairness Influencing to Consumer Loyalty. Journal of Shanxi Finance and Economics University.2010, 07, pp.94-103.
- [2] Zou Jianlan.Customer Loyalty Cultivation Based On Customer Satisfaction J.Journal of Harbin University of Commerce, 2010, 04, pp.58-60.