

Craft Payment Bussiness Requirement

- 1 [About the Project](#)
- 2 [Definitions](#)
- 3 [Introduction](#)
 - 3.1 [Document Overview](#)
- 4 [Bussiness Requirements Summary](#)
 - 4.1 [Bussiness Goals and Objectives](#)
 - 4.2 [Functional Requirements](#)
 - 4.3 [Non - Functional Requirements](#)
- 5 [System Analysis](#)
 - 5.1 [Existing system/Traditional Methods](#)
 - 5.2 [Proposed System](#)
 - 5.2.1 [Feasibiilty Study](#)
 - 5.2.1.1 [Technical Feasibility](#)
 - 5.2.1.2 [Economic Feasibility](#)
 - 5.2.1.3 [Behavioral Feasibility](#)
- 6 [Software development life-cycle\(SDLC\)](#)
- 7 [Design](#)
 - 7.1 [Logical Design](#)
 - 7.2 [Physical Design](#)
 - 7.3 [Design/Specification activities](#)
 - 7.4 [Input Design](#)
 - 7.4.1 [OBJECTIVES](#)
 - 7.5 [Output Design](#)
 - 7.6 [Data Flow Diagram](#)
 - 7.6.1 [0-LEVEL DFD](#)
 - 7.6.2 [1-LEVEL DFD](#)
 - 7.6.3 [Flow Chart](#)
- 8 [Database Design](#)
 - 8.1 [RELATIONAL DATABASE MANAGEMENT SYSTEM \(RDBMS\)](#)
 - 8.1.1 [RELATIONS, DOMAINS & ATTRIBUTES](#)
 - 8.1.2 [RELATIONSHIPS](#)
 - 8.1.3 [NORMALIZATION](#)
 - 8.1.3.1 [First Normal Form](#)
 - 8.1.3.2 [Second Normal Form](#)
 - 8.1.3.3 [Third Normal Form](#)
 - 8.2 [Tables Structure](#)
- 9 [Tools Used](#)
 - 9.1 [MySQL](#)
 - 9.2 [Mule ESB](#)
 - 9.2.1 [Login Flow](#)
 - 9.2.2 [Registration Flow](#)
 - 9.2.3 [Payment Flow](#)
 - 9.3 [Salesforce CRM](#)
 - 9.3.1 [Buyers Object](#)
 - 9.3.2 [CraftLeads Object](#)
 - 9.4 [CloudHub API](#)
 - 9.5 [Git and GitHub](#)
 - 9.6 [Jenkins](#)
 - 9.7 [Docker](#)
 - 9.8 [Docker Hub](#)
- 10 [Diagrams](#)
 - 10.1 [High Level Diagram](#)
 - 10.2 [Sequence Diagram](#)
 - 10.2.1 [Authentication Sequence Diagram](#)
 - 10.2.2 [Payment Sequence Diagram](#)

About the Project

The project titled as “CRAFT PAYMENT API “is a rest ful API. It is a payment gateway API integrated with existing digital processes can connect a checkout system to a payment network. It is online system that supports payment can pay bills for products and services online. Our Api provides such type of facilities for this systems.

Definitions

The following terms and abbreviations are used in this document:

Term	Definition
BRD	Business Requirements Document
24*7	Always
SDLC	Software Development Life Cycle
DevOps	Development and Operations
DFD	Data Flow Diagram
DBMS	Database Management System
RDBMS	Relational Database Management System
1NF	First Normal Form
PRI	Primary Key
UNI	Unique
FK	Foreign Key
ER Diagram	Entity Relationship Diagram
MySQL	My Structured Query Language
ESB	Enterprise Service Bus
JMS	Java Message Service
JDBC	Java Database Connectivity
HTTP	Hypertext Transfer Protocol
API	Application Programming Interface
RAML	RESTful API Modeling Language
CRM	Customer Relationship Management
CICD	Continous Integration, Contiouns Delivery
Etc	Et cetera

Introduction

The purpose of this Business Requirements Document (BRD) is to describe the different requirements for the Payment system built by craft. This document describes what the system would look like from a business perspective and lists critical requirements accurately in a technology-independent manner. Specifically, this document intends to describe;

- **Business Requirements** – high level business needs and goal of Online Payment
- **User Requirements** – needs of primary users who will be using the platform regularly
- **Non-Functional Requirements** – usability and behaviour characteristic required from the platform (in relation to things like the user interface, access security, availability, robustness, system failure, integration)

Document Overview

The main objective in this project is to create an online api that enables a company to sell a product and the buyer buys the product using an online payment system. Online payment has never been so easy. It is not restricted to just one system/company, it can be an api for all systems that support payment.

Bussiness Requirements Summary

Bussiness Goals and Objectives

Online Payment Systems are known to have the capability of replacing the traditional way of payment process. It is a term used for performing balance checks, account transactions, payments etc, via an online website. Craft payment is one this type of systems. It provides a payment api for online systems that supports this type of system. for It is easy to be skeptical about online payment systems. Security and privacy are critical for electronic communication and e-bussiness.

Functional Requirements

The requirements listed in the document must be business-related, relevant, necessary, prioritized, and meet the **SMART** criteria:

Specific: Precise and detailed, without ambiguity and using consistent terminology

Measurable: Testable and/or verifiable

Achievable: Technically feasible

Realizable: Realistic, given the resources

Traceable: Linked to Subsequent requirements

=> Customers:

- Can create account
- Can login with their credential
- Can view their account status
- Can pay using the system

Non - Functional Requirements

- Secure access of confidential data
- 24*7 availability
- Browse testing and support for Edge, Mozilla, Opera and Chrome
- Can integrate any system that support online payment system
- Scalability requirements
- Extendibility requirements
- Monitoring requirements
- Latency requirements
- Regulatory compliance requirements
- Data privacy requirements
- Auditing requirements
- Configurability requirements
- Performance requirements

System Analysis

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

Existing system/Traditional Methods

In traditional payment systems, a customer can open any bank account in banks, take the facility of saving his money by depositing money in local bank and pay through this account. Or he go to the store in person and pay using credit card, in cash or by other manual means.

Disadvantages of Traditional systems:

- Robbery:

Robbery is a big disadvantage these days. No one do the same thing in e-payment.

- Timing:

Stores and Banks usually are open in day time. And they are not opened all the time. But it may be possible that we have done payment in all the time. It is easy through e-payment rather than traditional way.

Problem Definition

- To allow only authorized user to access various functions and processes available in the system.
- Reduced clerical work as most of the work done by computer.
- Provide greater speed and reduced time consumption.
- To increase the number of customers for one company.
- This will reduce the manual workload and give information instantly.
- The software will maintain the list of accounts and customer record and balance status.
- The app will be user friendly, easily integrated with other systems so that even a beginner can operate the package and thus maintain the status of account, balance and payment log data status easily.

Proposed System

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work. The existing system has several disadvantages and many more difficulties to work with. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system will help the user to reduce the workload and mental conflict. The proposed system helps the user to work user friendly and he can easily do his jobs without time lagging.

Advantages of our system:

Access your account anytime, anywhere. While online payment you can securely pay anytime and anywhere you want as long as you have access to the Internet. You can use your computer - or, with some bank, even your smartphone - to pay bills on the go.

Convenience: This is the single most important benefit that outweighs any shortcoming of internet payment. Making payments right from the comfort of home or office at the click of a button without even having to step out is a facility none would like to forego. Keeping a track of accounts through the internet is much faster and convenient as compared to going to the store and the bank for the same.

Better Rates: The stores and online systems stand to gain significantly by the use of online payment system as it implies lesser physical effort from their end. The need to acquire larger spaces for offices and employ more staff to deal with the customers is significantly reduced making it financially beneficial to them.

Services: Technology has made it extremely convenient for the bank and stores as well as the customer to access to a host of wonderful services by simply logging in. These services include financial planning capabilities, functional budgeting and forecasting tools.

Mobility: Internet payment has a step further in the last few years in the form of mobile internet banking which accords unlimited mobility to the customer who can handle financial transactions even while on the move.

Another important benefit of the concept of online payment is that it is good for the environment as it cuts down the usage of paper, reduces pollution as people do not have to travel physically and also does not add emissions.

So the proposed system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations.

Feasibility Study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

Ø Does the existing technology sufficient for the suggested one?

Ø Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

Economic Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

Ø The costs conduct a full system investigation.

Ø The cost of the hardware and software.

Ø The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

Behavioral Feasibility

This includes the following questions:

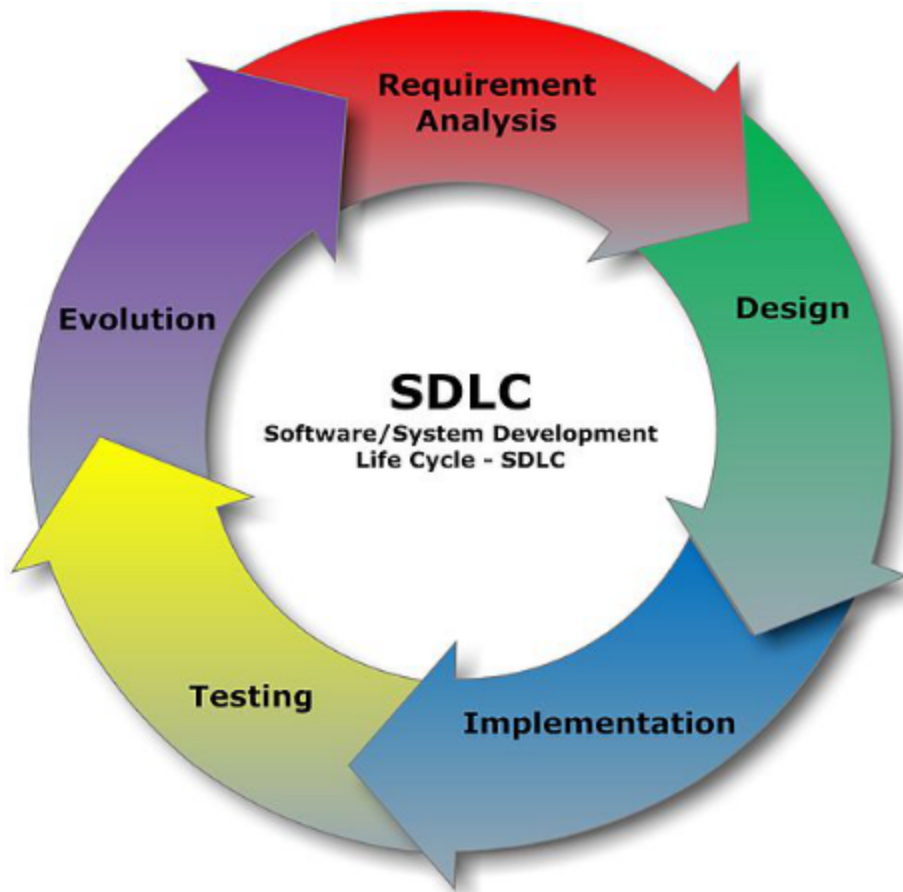
Ø Is there sufficient support for the users?

Ø Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

Software development life-cycle(SDLC)

The systems development life cycle (SDLC), or software development process in systemm engineering, information systems and software engineering, is a process of creating or altering information systems, and the models and methodologies that people use to develop these systems. In software engineering, the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system: the software development process:



A Systems Development Life Cycle (SDLC) adheres to important phases that are essential for developers, such as planning, analysis, design, and implementation, and are explained in the section below. It includes evaluation of the present system, information gathering, feasibility study, and request approval. A number of system development life cycle (SDLC) models have been created: waterfall, fountain, spiral, build and fix, rapid prototyping, incremental, and synchronize and stabilize. The oldest of these, and the best known, is the waterfall model: a sequence of stages in which the output of each stage becomes the input for the next. These stages can be characterized and divided up in different ways, including the following. Our project uses a DevOps approach, where the development team and the operation team collaborate together on the software lifecycle:

- **Systems analysis, requirements definition:** Defines project goals into defined functions and operation of the intended application. Analyzes end-user information needs.
- **Systems design:** Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudocode and other documentation.
- **Development:** The real code is written here.
- **Integration and testing:** Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.
- **Acceptance, installation, deployment:** The final stage of initial development, where the software is put into production and runs actual business.
- **Maintenance:** What happens during the rest of the software's life: changes, correction, additions, moves to a different computing platform and more. This is often the longest of the stages.



Design

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

Logical Design

The logical flow of a system and define the boundaries of a system. It includes the following steps:

- Ø Reviews the current physical system – its data flows, file content, volumes , Frequencies etc.
- Ø Prepares output specifications – that is, determines the format, content and Frequency of reports.
- Ø Prepares input specifications – format, content and most of the input functions.
- Ø Prepares edit, security and control specifications.
- Ø Specifies the implementation plan.
- Ø Prepares a logical design walk through of the information flow, output, input, Controls and implementation plan.
- Ø Reviews benefits, costs, target dates and system constraints.

Physical Design

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

- Ø Design the physical system.
- Ø Specify input and output media.
- Ø Design the database and specify backup procedures.
- Ø Design physical information flow through the system and a physical design
- Ø Plan system implementation.
- Ø Prepare a conversion schedule and target date.
- Ø Determine training procedures, courses and timetable.
- Ø Devise a test and implementation plan and specify any new hardware/software.
- Ø Update benefits , costs , conversion date and system constraints

Design/Specification activities

- Ø Concept formulation.
- Ø Problem understanding.
- Ø High level requirements proposals.
- Ø Feasibility study.
- Ø Requirements engineering.
- Ø Architectural design.

5.3 Module Design

Customer Module:

- Can create account

- Can login with his/her credential
- Can process payment
- Can access his account status

Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow
- The api provides a secure resource for each system.

Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

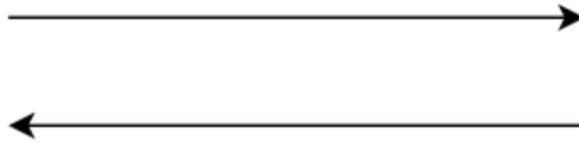
- o Convey information about past activities, current status or projections of the Future.
- o Signal important events, opportunities, problems, or warnings.
- o Trigger an action.
- o Confirm an action.

Data Flow Diagram

A **data flow diagram (DFD)** is a graphical representation of the "flow" of data through an information system, modeling its *process* aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

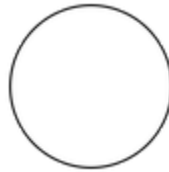
A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).the basic notation used to create a DFD's are as follows:

1. **DataFlow:** data moves in a specific from an origin to a destination.



2. Process:

People ,procedures or device that use or produce data. The physical components not identified.



3. **Source:** external sorce or destination of data, which may be people programs, organizations or other entities.

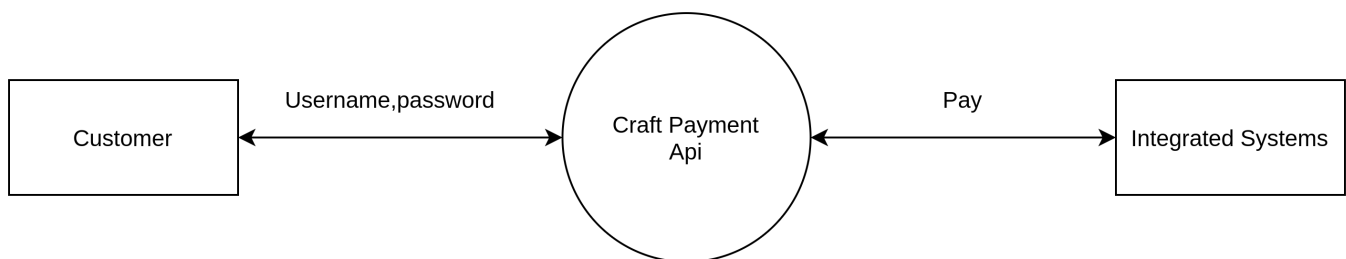


4 .**Data source:** here data are store and referenced by a process in the system.



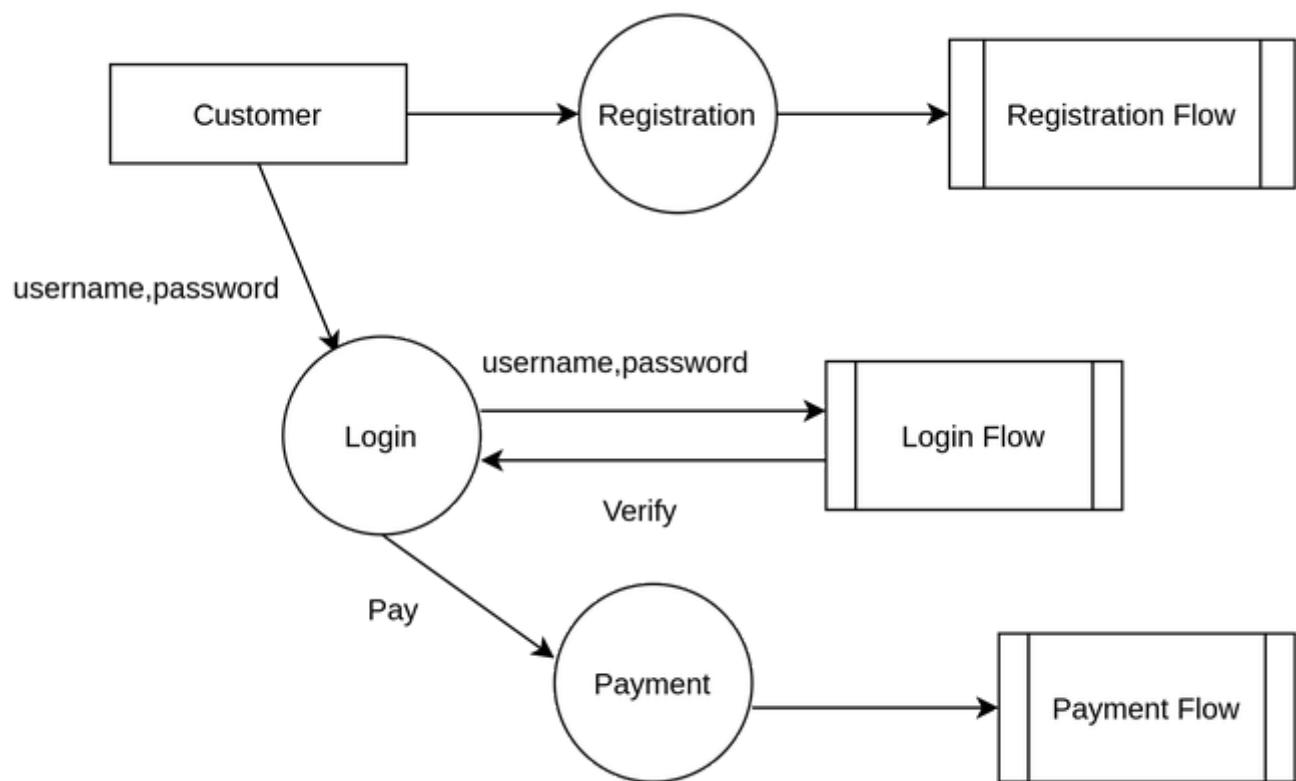
0-LEVEL DFD

For the customer:



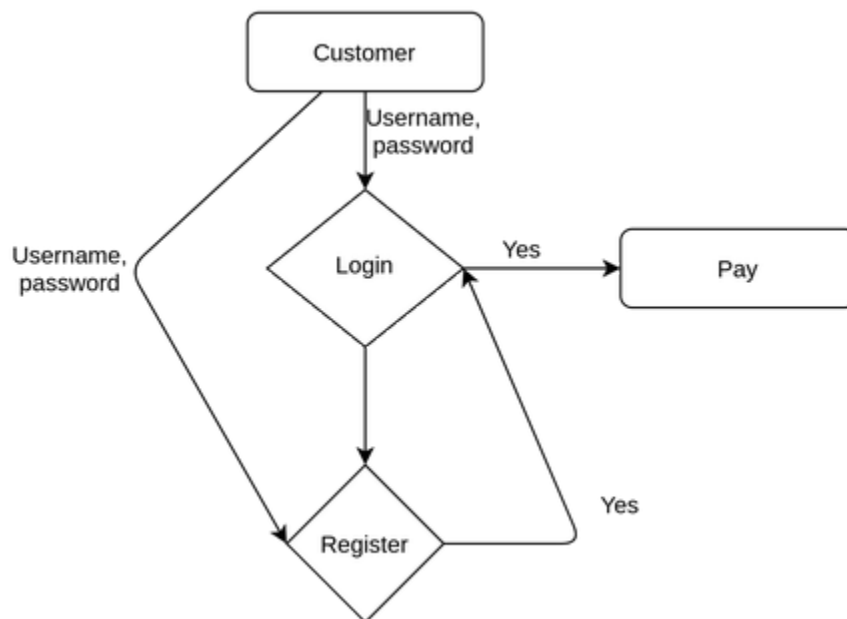
1-LEVEL DFD

For the customer:



Flow Chart

For the customer



Database Design

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- o Data Integrity

- o Data independence

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this system for the following reasons.

- To structure the data so that there is no repetition of data , this helps in saving.
- To permit simple retrieval of data in response to query and report request.
- To simplify the maintenance of the data through updates, insertions, Deletions.
- To reduce the need to restructure or reorganize data which new application Requirements arise.

RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a table represents a set of related values.

RELATIONS, DOMAINS & ATTRIBUTES

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

RELATIONSHIPS

- o Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- o Entity Integrity enforces that no Primary Key can have null values.
- o Referential Integrity enforces that no Primary Key can have null values.
- o Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.
- o Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity.

NORMALIZATION

As the name implies, it denoted putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These includes:

- ü Normalize the data.
- ü Choose proper names for the tables and columns.
- ü Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows "relations within relations" or "relations as attribute values within tuples". The only attribute values permitted by 1NF are single atomic or indivisible values.

The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each nonatomic attribute or nested relation. This eliminated repeating groups of data.

A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no nonkey attribute should be functionally dependent on a part of the primary key.

In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependant on a part of the key.

A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non key attribute functionally determined by another non key attribute or by a set of non key attributes. That is, there should be no transitive dependency on the primary key.

In this we decompose and set up relation that includes the non key attributes that functionally determines other non key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key.

A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non key attribute.

Tables Structure

Table Name: user_info

Field	Type	Null	Key	Default	Extra
id	int unsigned	NO	PRI	NULL	auto_increment
status	enum('active','inactive')	YES	-	active	-
registration_date	timestamp	YES	-	CURRENT_TIMESTAMP	DEFAULT_GENERATED
email	varchar(255)	NO	UNI	NULL	-
card_no	varchar(500)	YES	-	NULL	-
password	varchar(500)	YES	-	NULL	-

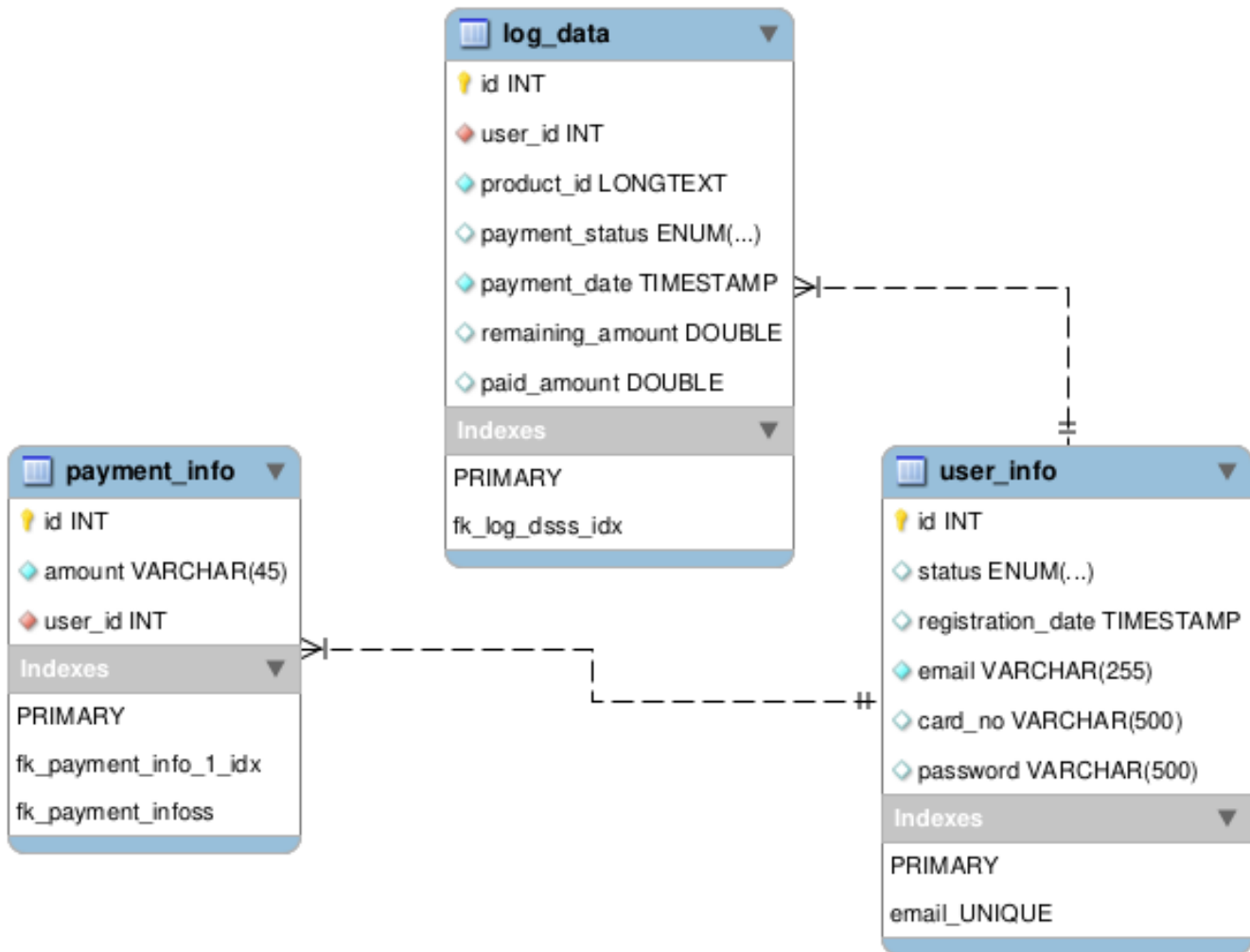
Table Name: payment_info

Field	Type	Null	Key	Default	Extra
id	int unsigned	NO	PRI	NULL	auto_increment
amount	varchar(45)	NO	-	NULL	
user_id	int unsigned	NO	FK	NULL	

Table Name: log_data

Field	Type	Null	Key	Default	Extra
id	int unsigned	NO	PRI	NULL	auto_increment
user_id	int unsigned	NO	FK	NULL	-
product_id	longtext	NO	-	NULL	-
payment_status	enum('active','inactive')	YES	-	active	-
payment_date	timestamp	NO	-	CURRENT_TIMESTAMP	DEFAULT_GENERATED
remaining_amount	double	YES	-	NULL	
paid_amount	double	YES	-	NULL	

ER diagram for the database



Tools Used

MySQL

MySQL is the most popular Open Source Relation SQL database management system. MySQL Is one of the best RDBMS being used for developing software applications. The design of this database and its tables is mentioned in section 6 of this document (refer). We store payment data and user data in this database.

Mule ESB

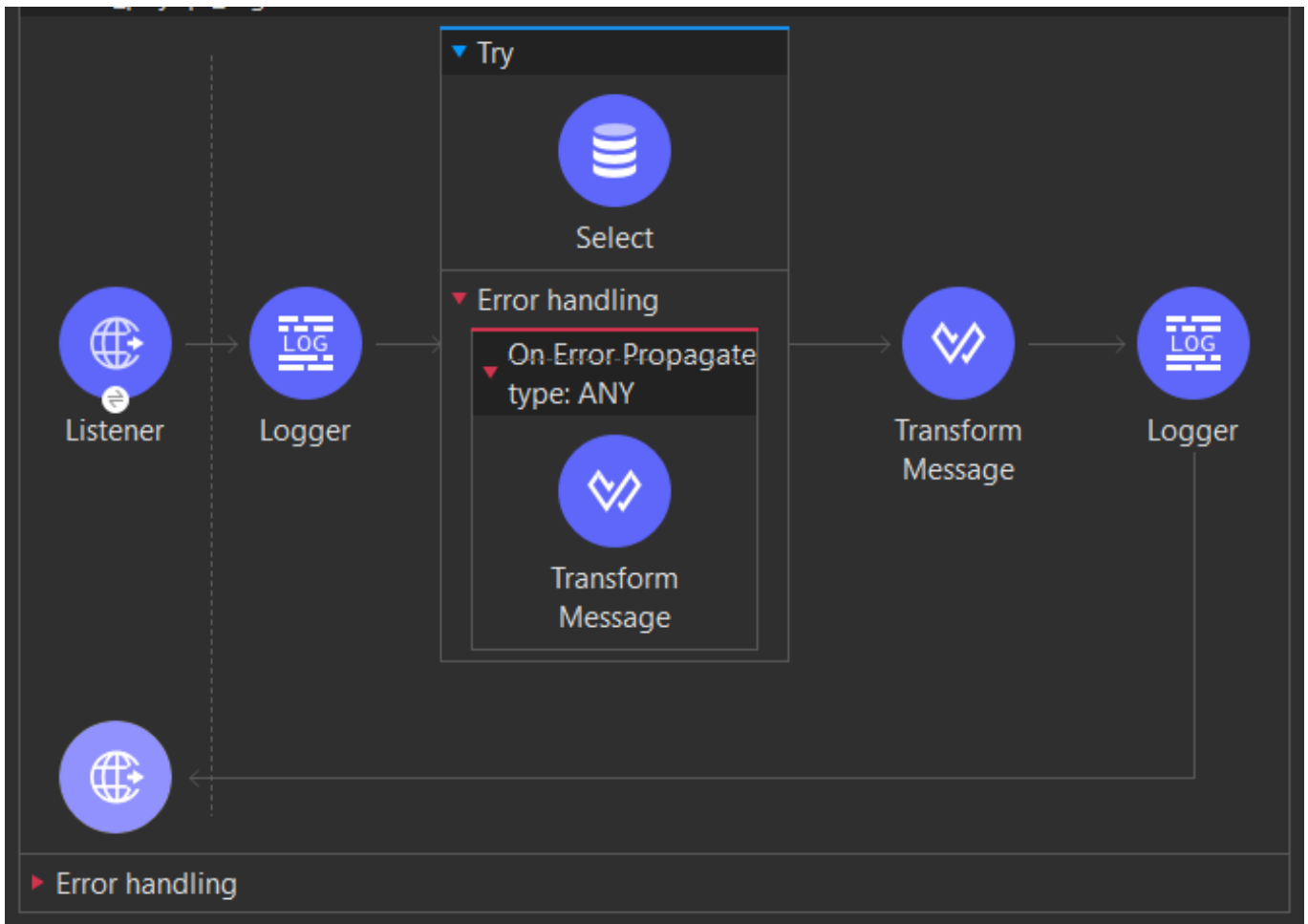
Mule, the runtime engine of Anypoint platform, is a lightweight Java-based enterprise service bus (ESB) and integration platform that allows developers connect applications together quickly and easily, enabling them to exchange data. It enables easy integration quickly and easily, enabling them to exchange data. It enables easy integration of existing systems, regardless of the different technologies that the applications use, including JMS, Web Services, JDBC, HTTP, and many more. The ESB can be deployed anywhere, can integrate and orchestrate events in real time or in batch and has universal connectivity. The key advantage of an ESB is that it allows different applications to communicate with each other by acting as a transit system for carrying data between applications within your enterprise or across the internet.

In mule, craft payment API we use **design first approach**. In this approach we first write a RAML for the api and create the flow. We use Mule for full api development and integration tool for the database and salesforce system.

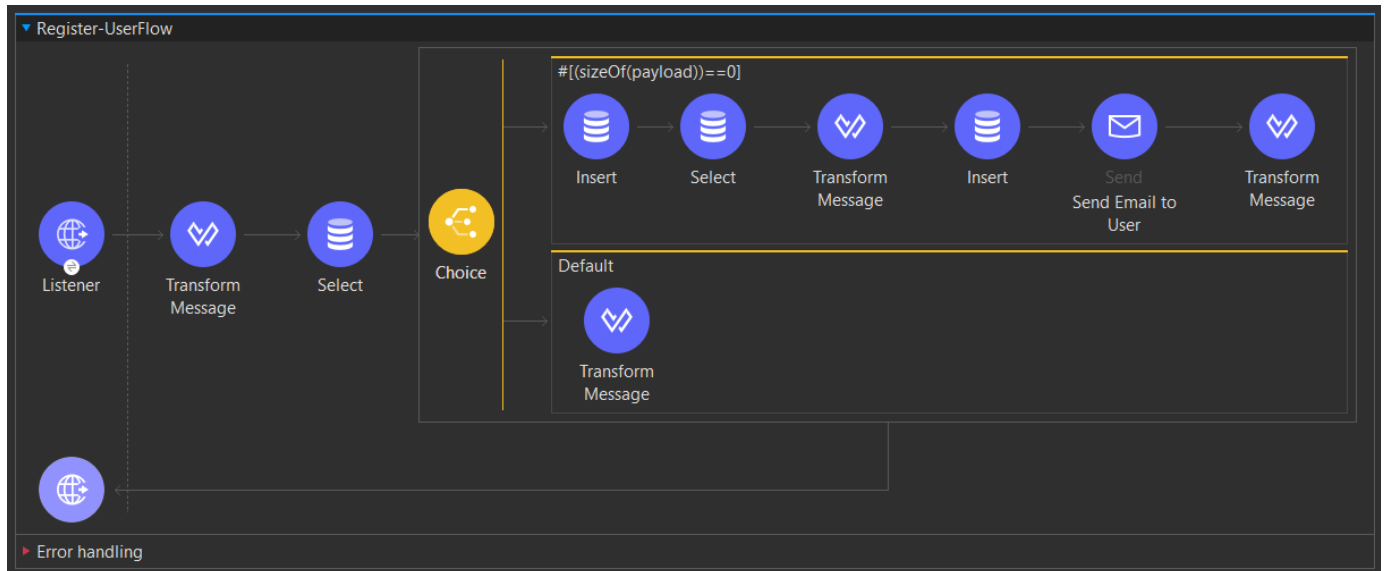
There are three main flows for each operation. Login flow, Registration flow and Payment flow.

Login Flow

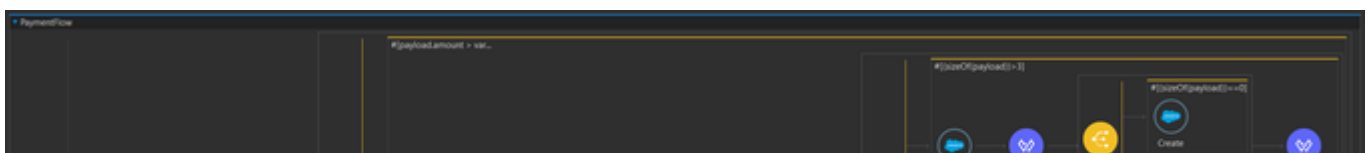
▼ craft pavapi loginFlow

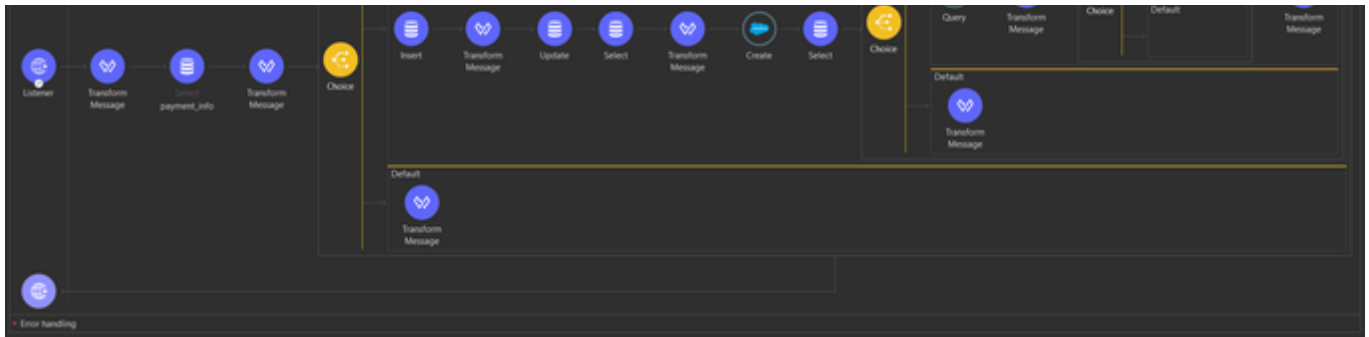


Registration Flow



Payment Flow





Salesforce CRM

Salesforce CRM is a cloud-based software that helps organizations to effectively streamline their sales and marketing operations. Today, Salesforce is the #1 CRM platform in the world. It also offers organizations easy access to web based software over the internet.

We use salesforce to store customer as lead and opportunity. In our case any customer that pays using our platform (api) stored in opportunity and if that customer data stored morethan three times that customer is stored in lead. For this purpose we created **Buyers** custom object for the opportunity and **CraftLeads** for the lead

Buyers Object

Action	Buyer Name	Email	CardNumber	Created By	Created Date
<input type="checkbox"/> Edit Del	a000900000GUBLN	michaeltaye0123@gmail.com	123456	Craft Payment, 29/09/2021, 12:31	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUBLS	michaeltaye0123@gmail.com	123456	Craft Payment, 29/09/2021, 12:32	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUBQ0	michaeltaye0123@gmail.com	123456	Craft Payment, 29/09/2021, 12:54	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUBQ5	michaeltaye0123@gmail.com	123456	Craft Payment, 29/09/2021, 12:55	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUBQ7	michaeltaye0123@gmail.com	123456	Craft Payment, 29/09/2021, 14:01	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUBQk	michaeltaye0123@gmail.com	123456	Craft Payment, 29/09/2021, 13:24	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUBul	michaeltaye0123@gmail.com	123456	Craft Payment, 29/09/2021, 13:28	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUC4z	michaeltaye0123@gmail.com	123456	Craft Payment, 29/09/2021, 13:31	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUC59	michaeltaye0123@gmail.com	123456	Craft Payment, 29/09/2021, 13:33	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUCC0	michaeltaye0123@gmail.com	123456	Craft Payment, 29/09/2021, 13:38	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUCIM	michaeltaye0123@gmail.com	123456	Craft Payment, 29/09/2021, 13:56	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUCmi	michaeltaye012@gmail.com	123456	Craft Payment, 29/09/2021, 14:05	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUCmx	michaeltaye012@gmail.com	123456	Craft Payment, 29/09/2021, 14:09	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUCn2	michaeltaye012@gmail.com	123456	Craft Payment, 29/09/2021, 14:10	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUCn7	michaeltaye012@gmail.com	123456	Craft Payment, 29/09/2021, 14:10	29/09/2021
<input type="checkbox"/> Edit Del	a000900000GUE3S	michaeltaye012@gmail.com	123456	Craft Payment, 29/09/2021, 14:50	29/09/2021

CraftLeads Object

Action	CraftLead Name	CardNumber	Email	Created By	Created Date
<input type="checkbox"/> Edit Del	a010900001Vk8d7	123456	micha@gmail.com	Craft Payment, 29/09/2021, 15:19	29/09/2021
<input type="checkbox"/> Edit Del	a010900001VkJQr	123456	michaeltaye0123@gmail.com	Craft Payment, 29/09/2021, 16:21	29/09/2021

CloudHub API

The CloudHub management API (CloudHub API) enables you to programmatically access the functions of Anypoint Runtime Manager, such as create, deploy, start, stop, list, and delete an application on CloudHub. We use CloudHub to deploy our API.

Git and GitHub

Git is a version control tool that allow you to perform all kinds of operations to fetch data from the central server or push data to it and GitHub is a core hosting platform for version control collaboration. GitHub is a company that allows you to host a central repository in a remote server. We use Github for to store our project and git for the version controlling.

Jenkins

Jenkins is an open source automation server. It helps automate the parts of the software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery (cicd). It is a server-based system built by java that runs in servlet containers such as Apache Tomcat. We use jenkins for continuous integration and continuous deployment strategy. Since Mule build the project using Maven so we integrate jenkins with maven, git and cloud hub for continuous deployment.

Docker

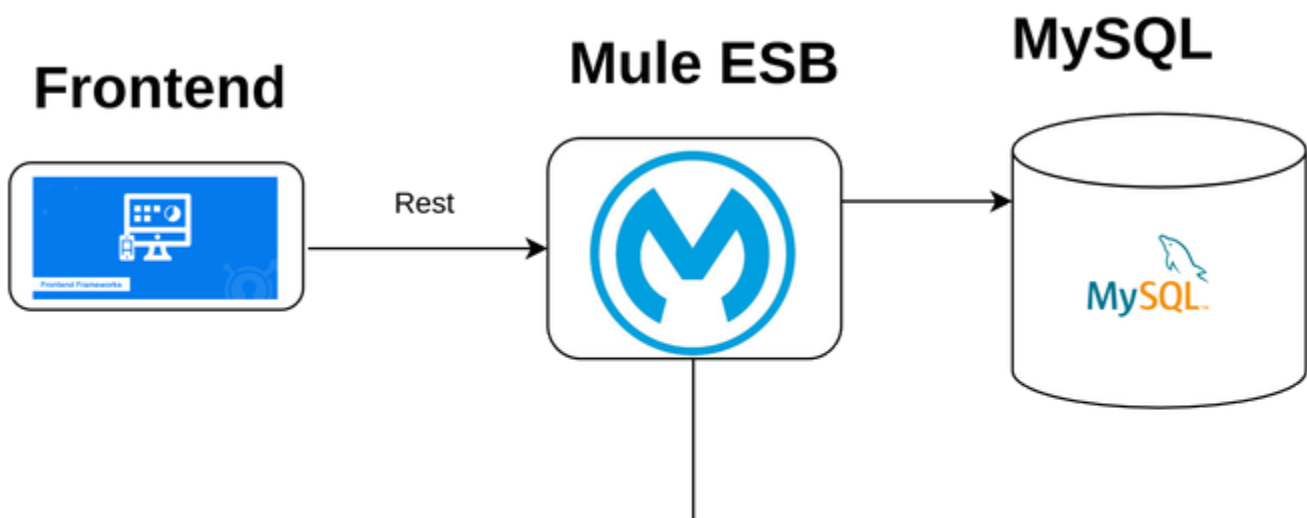
Docker is an open source platform for building, deploying and managing containerized applications. It enables developers to package applications into containers - standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. We use docker for building an image for our application.

Docker Hub

Docker Hub is a cloud-based repository in which Docker users and partners create, test, store and distribute container images. Through Docker Hub, a user can access public, open source image repositories, as well as use a space to create their own private repositories, automated build functions, and work groups. We use Docker Hub to store our build image.

Diagrams

High Level Diagram

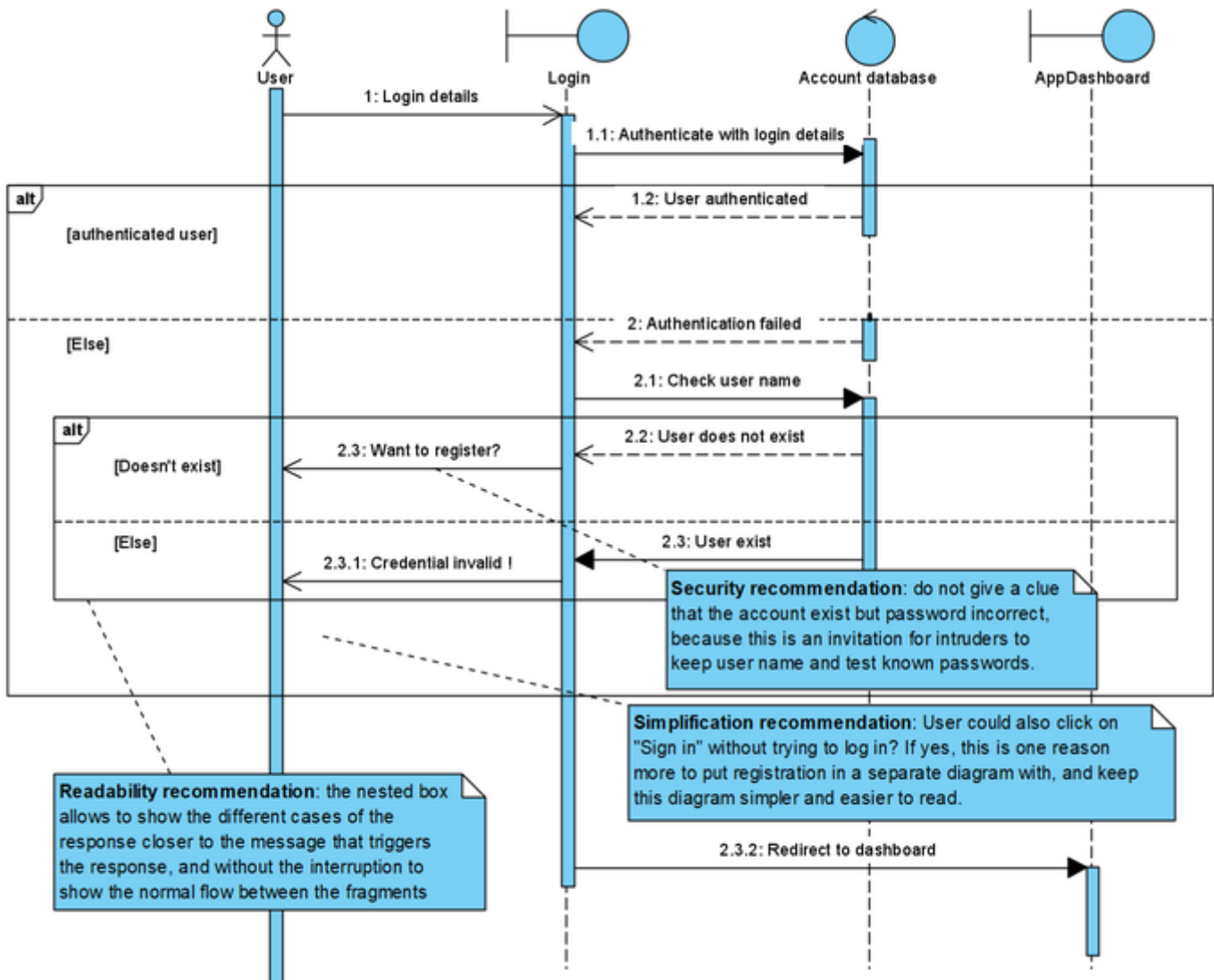


Salesforce



Sequence Diagram

Authentication Sequence Diagram



Payment Sequence Diagram

