High Level Design (HLD)
Customer Relationship Management

Revision Number: 1.0
Last date of revision: 02/06/2021

Mahendra Cherukupalli
Abe Alexander

Document Version Control

Date Issued	Version	Description	Author
01th June 2021	1.0	First Version of Complete HLD	

INDEX

ABSTRACT

Customer relationship management (CRM) is a necessity in any customer-focused industry. For banks, it's an especially useful tool for meeting sales and marketing goals and exceeding customer expectations. CRM concern the relation between the organizations along with its consumers.

This project aims to study why CRM is important in banks, how CRM works in banks, and also the effectiveness of clients in obtaining long-term customer relationships, customer loyalty, and customer satisfaction by the use of CRM. And also suggest feasible recommendations to clients to increase customer satisfaction and market share by the effective use of CRM.

The motivation behind using AI & Machine learning in CRM is AI can have the ability to measure customer behaviour and their expectations for the said product. CRM with artificial intelligence allows enterprises to combine all strategies and technologies to streamline, manage and analyse customer interactions. On top of that, machine learning technology can examine previous decisions to see which actions might lead to better decisions.

1. INTRODUCTION

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	Description
CRM	Customer Relationship Management
CLV	Customer Lifetime Value
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment
AWS	Amazon Web Services

2. GENERAL DESCRIPTION

2.1 Product Perspective:

The CRM Tool is a web application which will show customers segmented into various categories, predict customer churn and suggest possible direct marketing actions. SQL is used to retrieve, insert, delete, and update the database. Here the system store each and every data given by user or received in request to the MySQL/MongoDB database.

2.2 Tools used

Python programming language and frameworks such as Numpy, Pandas, Matplotlib, Seaborn. Scikit-learn, are used to build the whole model.

- Python is used as a programing language.
- Numpy, Pandas are used to perform data manipulation and analysis.
- Matplotlib, Seaborn are used for visualization of the plots.
- Scikit-learn is used for Machine learning algorithms.
- PyCharm is used as IDE.
- Front end development is done using HTML/CSS
- Django is used for backend development.
- AWS is used for deployment of the model.
- MongoDB is used to retrieve, insert, delete, and update the database.
- GitHub is used as version control system.
- Tableau/Power BI is used for dashboard creation.

2.3 Constraints

One of the greatest challenge's businesses face when implementing a CRM solution is cost. Yes, it might turn out as a costly affair if done in haste and without robust planning.

2.4 Assumptions

The ultimate purpose of CRM is to increase profit, which can be achieved mainly by providing a better service to customers than competitors by using Machine Learning techniques. It is also assumed that all aspects of this project have the ability to work together in the way the designer is expecting.

3. DESIGN DETAILS

3.1 Functional Architecture

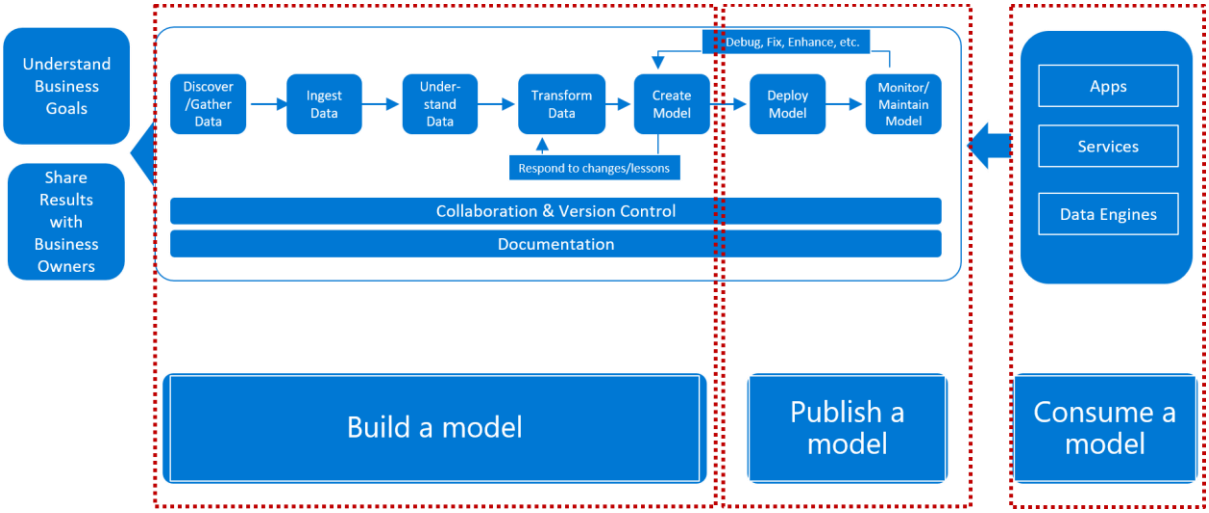


Figure 1: Functional Architecture of CRM

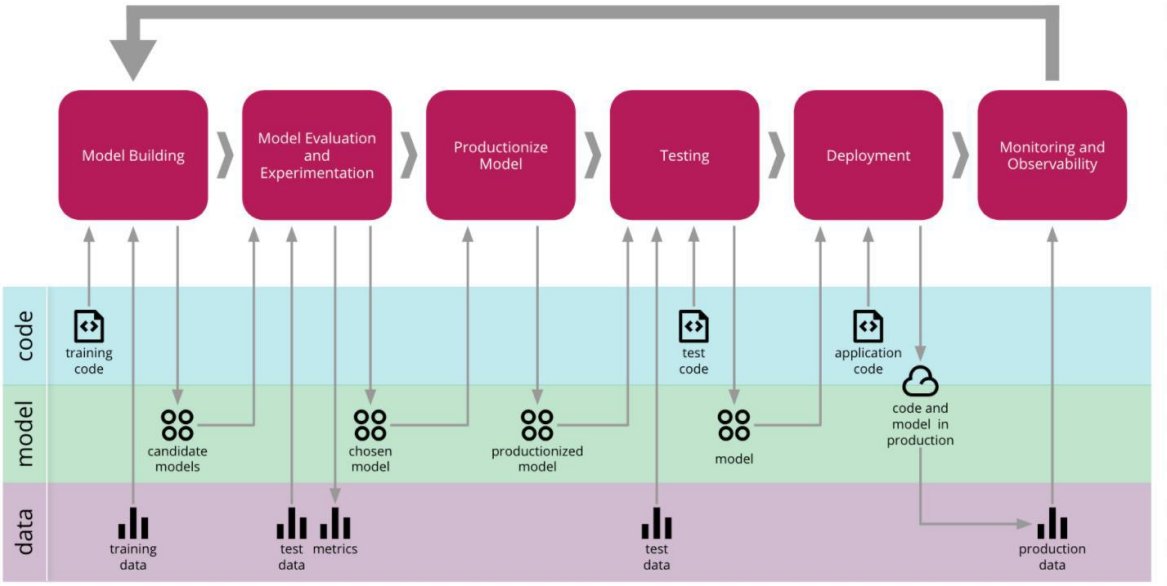


Figure 2: Continuous Delivery of CRM

3.2 Database Design

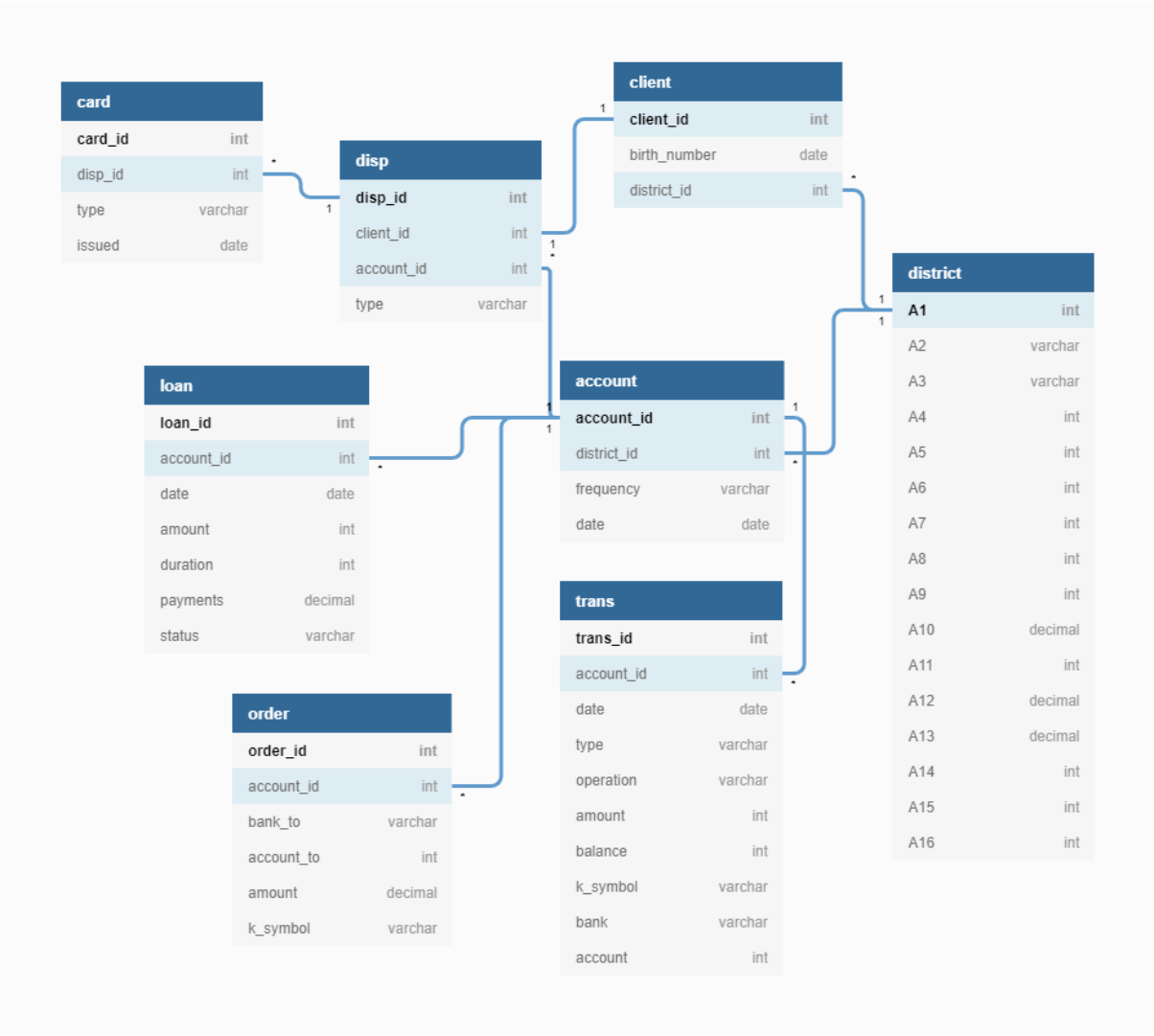


Figure 3: Database Design

3.3 Dashboard Application Architecture

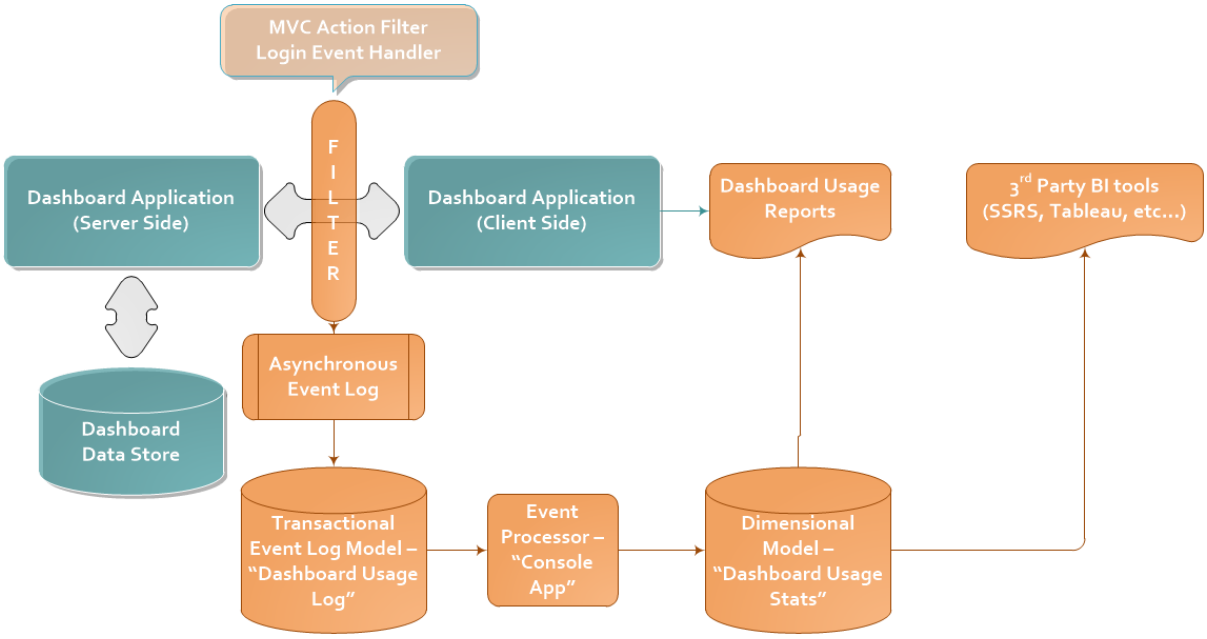


Figure 4: Dashboard application architecture

The user interface is a very simple plain layout with little graphics. It will display information very clearly for the user and will primarily output information to the user through Dashboards. Also, all the details for the user input will be provided.

3.4 Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.5 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

3.6 Help

The ‘Help’ option is provided in web application for guiding users on how to use the tool.

3.7 Performance

The CRM system is used for segmenting our customers into various segments based on which advertising budget and possible marketing actions can be performed so as to retain our high-level customers and spend less on our low-level customers. It should be as accurate as possible as the bank may lose out on potential revenue. Also, model retraining is very important to improve the performance.

3.8 Security

Since the CRM system consists of bank user’s data, the information should be secured.

3.9 Reusability

The code written and the components used should have the ability to be reused with no problems.

3.10 Application compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

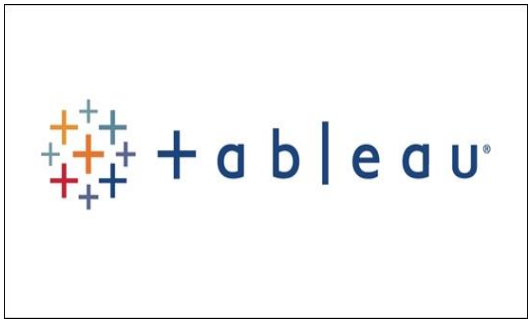
3.11 Resource utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

3.12 Deployment

4. Dashboards

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the CRM.



As and when, the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors

4.1 KPIs (Key Performance Indicators)

Key indicators displaying a summary of a customer as below:

1. Revenue by deposit interest, loan interest, service fees, and transaction fees.
2. Assets Under Management (AUM)
3. Client Survey Score
4. Upsell rate
5. Net-new revenue
6. Customer lifetime value (CLV)
7. Customer acquisition cost (CAC)
8. Revenue generated by campaign
9. Churn rate

5. REFERENCES