# Low Level Design (LLD)

# Bankbot

| | |
|---|---|
| Written By | Diana Laveena DSouza |
| Document Version | 0.1 |
| Last Revised Date | 13-Jul-2023 |

# Document Version Control

## Change Record:

| Version | Date | Author | Author |
|---|---|---|---|
| 0.1 | 13/07/2023 | Diana Laveena DSouza | Introduction & Architecture Defined |

## Reviews:

| Version | Date | Reviewer | Comments |
|---|---|---|---|
| 0.1 | 13/07/2023 | | Document Content, Version Control and Unit Test Cases to be added |

## Approval Status:

| Version | Review Date | Reviewed By | Approved By | Comments |
|---|---|---|---|---|
| | | | | |

# Contents

# Abstract

In today's world, online banking has become increasingly automated, but many people still encounter various issues during their online banking experience. Additionally, some individuals may not be aware of the correct safety measures to protect their accounts. To assist customers with their queries and provide guidance on online banking safety, a bot can be developed.

# 1 Introduction

## 1.1 Why is Low-Level Design Document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Facebook Status Prediction. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so the programmer can directly code the program from the document.

## 1.2 Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

# 2 Technical specifications

## 2.1 Dataset

| DataSet | Finalized | Source |
|---|---|---|
| intents | yes | https://github.com/IBM/watson-banking-chatbot/blob/master/data/conversation/workspaces/banking_IN.json |

### 2.1.1 Dataset Overview

The Dataset consists of a person's query(pattern) and tag.

**Dataset**

| tag | patterns |
|---|---|
| discoTerms | Can I select my EMI amount? |
| discoTerms | I have not received any letter asking my confirmation |
| discoTerms | Why I am not able to select loan tenure below 6 month? |
| discoTerms | If I Pay some Lumpsum amout ,will my EMI reduce? |
| discoTerms | Can I get credit of loan amount in Account other than mentioned on the screen |
| discoTerms | Can I enter a mobile number other than the one used for downloading the app. |
| discoTerms | Can I avail 2 Pre-approved Loans |
| discoTerms | Can FATCA status be updated later? |
| discoTerms | Is the Interest rate fixed or floating? |
| discoTerms | Why I am not able to select loan amount below 25000? |
| discoTerms | Which Branch do I have to go for signing the documents? |
| discoTerms | What if I want loan for higher amount and longer duration? |
| discoTerms | What happens if there is a spelling mistake in my name printed on Adhaar card? |
| discoTerms | My mobilebanking app is not working . What to do? |
| Greetings | Hi how are you |
| Greetings | how are you |
| Greetings | Hi |
| Greetings | Hello |
| Greetings | Howdy |
| Greetings | Good Day |
| recommend | looking to purchase |
| recommend | suggest me some gifts |
| recommend | want to buy |

### 2.1.2 Input schema

| Feature Name | Datatype | Size | Null/Required |
|---|---|---|---|
| patterns | text | 250 | Not Required |
| tags | varchar | 20 | Not Required |

## 2.2 Predicting tags

- The system presents the set of inputs required from the user.
- The user gives the required information.
- The System should be able to predict tags.

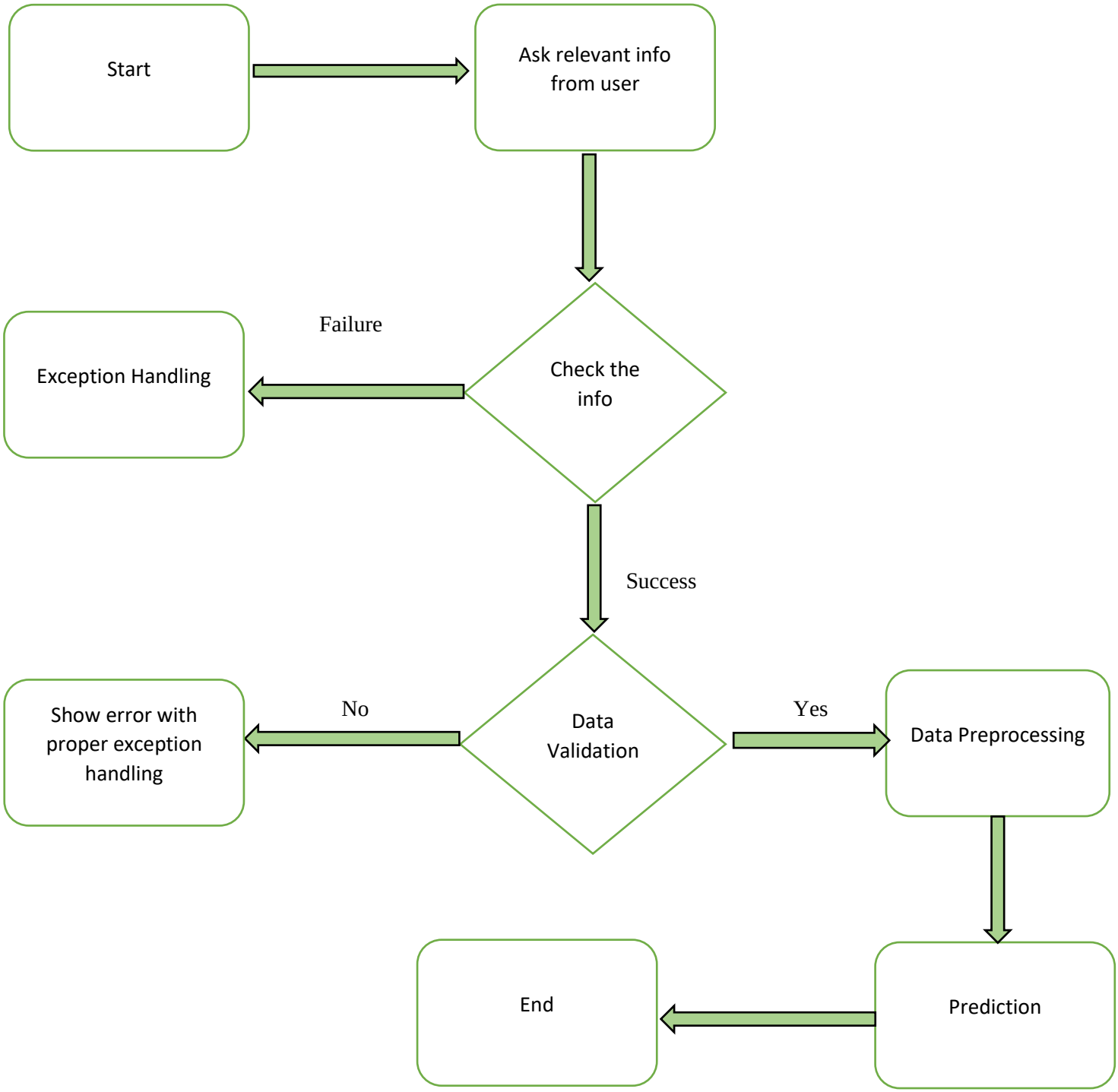
## 2.3 Deployment

# 3 Technology Stack

| | |
|---|---|
| Front End | HTML/CSS |
| Backend | Python Flask |
| Database | Mysql |
| Deployment | Android, Local Server |

# 4 Model Training/Validation Workflow

# 5 User I/O workflow



# 6 Test Cases

| Test Case Description | Pre-Requisite | Expected Result |
|---|---|---|
| Verify whether the Application URL is accessible to the user | 1. Application URL should be defined | The application URL should be accessible to the user. |
| Verify whether the Application loads entirely for the user when the URL is accessed | 1. Application URL is accessible<br>2. Application is deployed | The Application should load entirely for the user when the URL is accessed. |
| Verify whether the user can input the text in all input fields | 1. Application is accessible | The user should be able to input the text in all input fields. |
| Verify whether the user gets Submit button to submit the inputs. | | The user should get Submit button to submit the inputs. |
| Verify whether the user is presented with results on clicking submit. | | The user should be presented with results on clicking submit |