# Low Level Design (LLD)

# Bankbot

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

# Document 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

# 1 Introduction

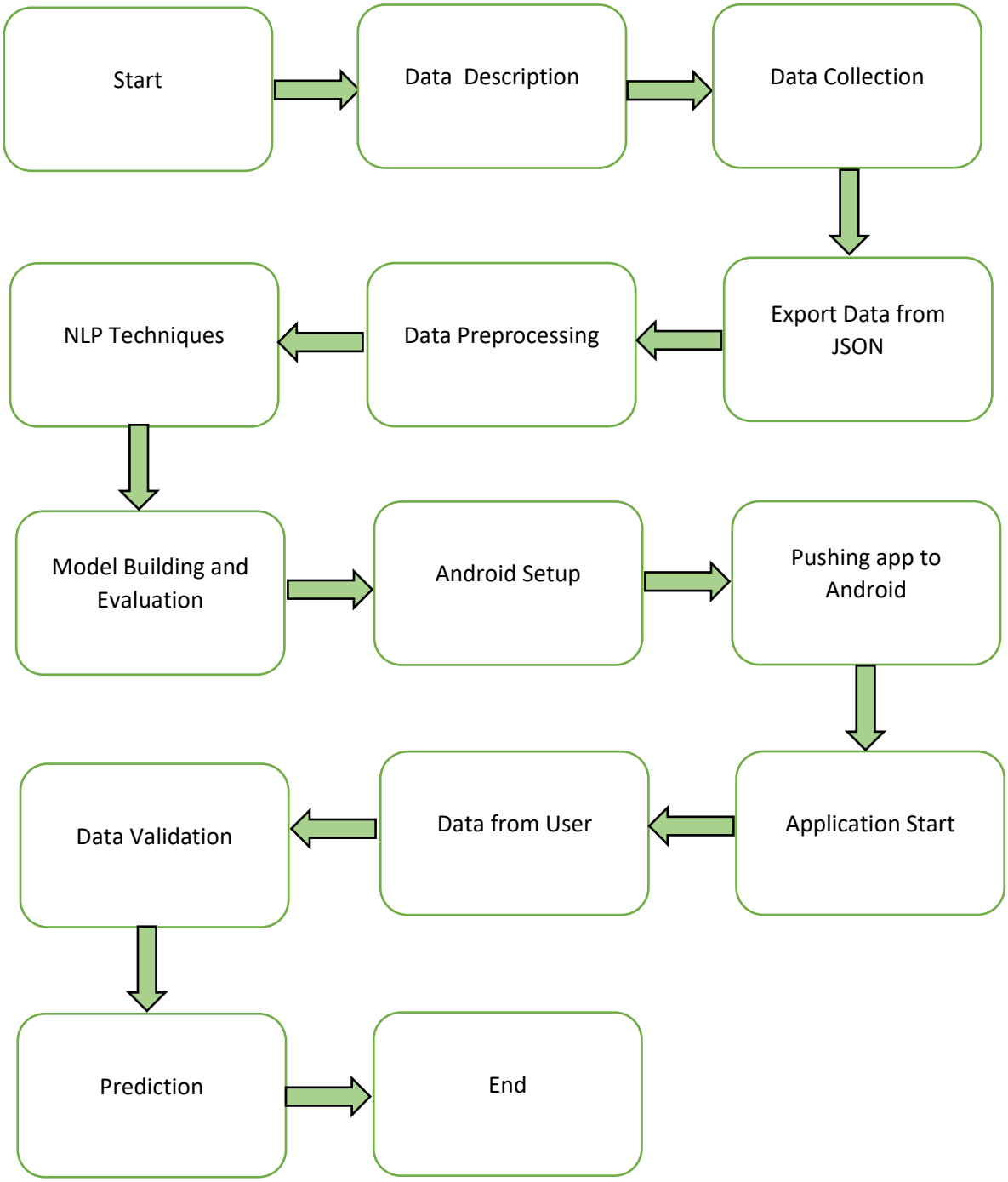
## 1.1 Why is Low-Level Design Document?

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 Architecture



# 3 Architecture Description

## 3.1 Data Description

**Context**: Collection of patterns and tags helps greatly in NLP Classification tasks

**Content:** List of intents with patterns; tags and responses for building the deep learning model.

## 3.2 Data Collection

Data Collected from the Github.

## 3.4 Export Data from Database

Data Export from Database – The data in a JSON is exported as a CSV file to be used for Text Pre-processing and Model Training.

## 3.5 Data Preprocessing

Data Preprocessing steps, we could use stop word removal, punctuation removal, tokenization, stemming, TFIDF, Label Encoder etc.

## 3.6 Model Building and Evaluation

We will find the best model is used to identify different tags based on accuracy.

## 3.7 Data from User

Here we will collect data from users.

## 3.8 Data Validation

Here Data Validation will be done, given by the user.

## 3.9 Prediction

The model will predict the tags and the tag response will be the answers to the customer queries. Additionally, it will give transaction details, security changes based on tags.

.

## 3.10 Deployment

We will be deploying the model to local server. The responses will be displayed in Android app.

This is a workflow diagram for the Bankbot Prediction.

# 4 Unit 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 |