**COMSATS University Islamabad,   
Park Road, Chak Shahzad, Islamabad Pakistan**

SOFTWARE DESIGN DESCRIPTION   
(SDD DOCUMENT)

for

**Banking Bot**  
Version 1.0

***By***

**Muhammad Usman CUI/SP15-BSE-079/ISB**

**Hamza Shabbir CUI/FA15-BSE-065/ISB**

***Supervisor*Dr. Sohail Asghar**

*Bachelor of Science in Software Engineering*

**Table of Contents**

[1. Introduction 1](#_Toc532199904)

[2. Design methodology and software process model 1](#_Toc532199905)

[3. System overview 1](#_Toc532199906)

[3.1 Architectural design 1](#_Toc532199907)

[3.2 Process flow/Representation 2](#_Toc532199908)

[4. Design models 8](#_Toc532199909)

[5. Data design 11](#_Toc532199910)

[5.1 Data dictionary 11](#_Toc532199911)

[6. Algorithm & Implementation 11](#_Toc532199912)

[7. Software requirements traceability matrix 11](#_Toc532199913)

[8. Human interface design 12](#_Toc532199914)

[8.1 Screen images 12](#_Toc532199915)

[8.2 Screen objects and actions 12](#_Toc532199916)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

Supervised by

<Supervisor’s Name>

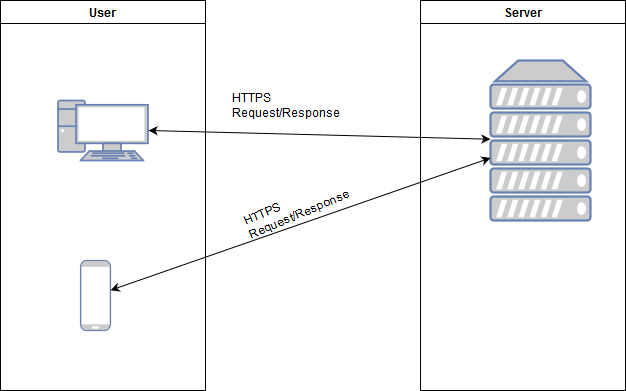
Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Introduction

The proposed system is an artificially intelligent bot designed for a particular bank for eg (HBL). This system analyze and understands human language and responds according to their query. There is no specific format of asking a query. As of now, users can ask questions related to various bank activities like debit card and Home Loan.

# Design methodology and software process model

This system uses procedural design methodology. The construction of this system requires intense deep learning models which therefore is not based on object oriented paradigm.

The selected software process for the proposed system is agile method. In agile method each module is developed in the form of small increments. The rationale behind using agile methodology for this system is to anticipate changes without making huge modifications to schedule. Secondly, this process methodology entails breaking down project into prioritized requirements. This means we can develop and test the requirement with the highest priority first.

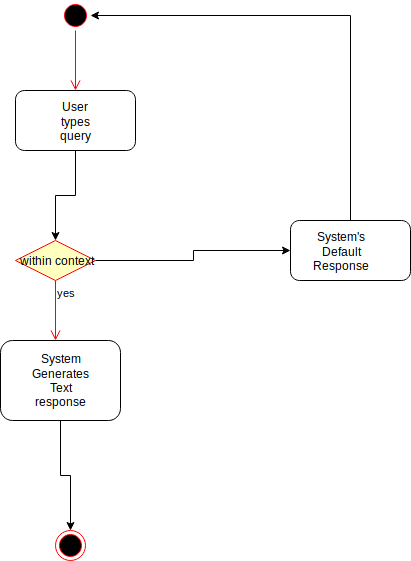
# System overview

## Architectural design

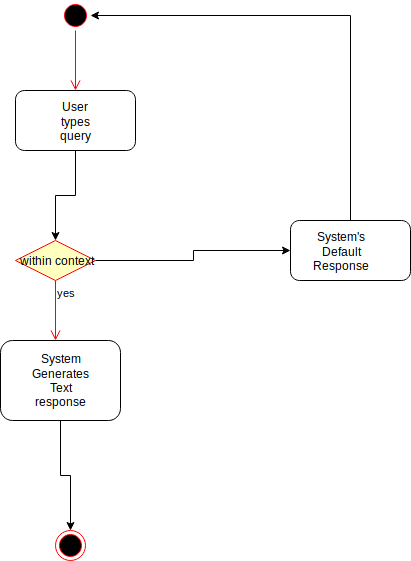
Our architectural design model is Client Server Architecture Model

**Figure 1: Client Server Architecture**

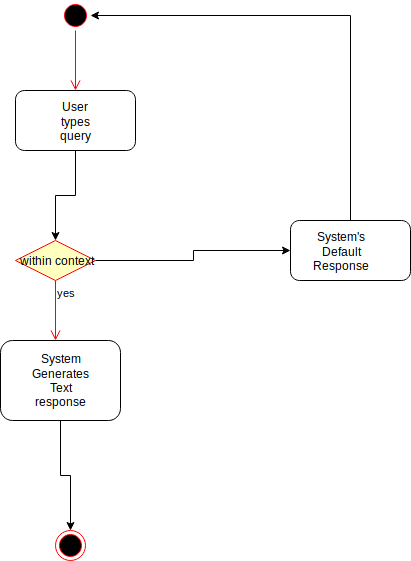
## Process flow/Representation



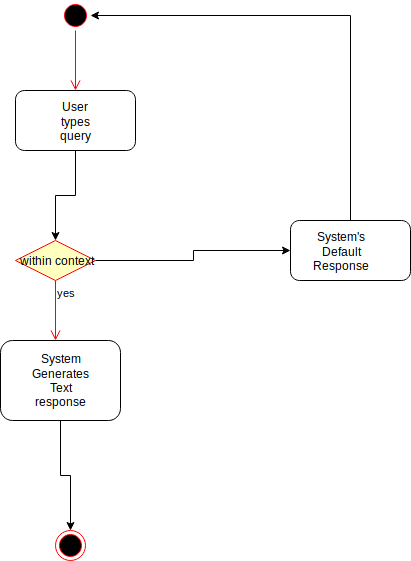
**Figure 2: ATM Query Processing**



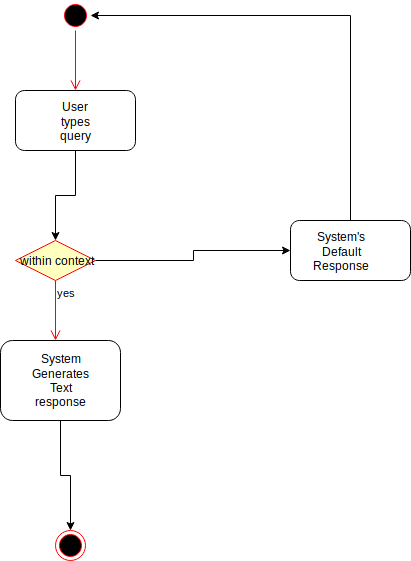
**Figure 3: Home loan Query Processing**



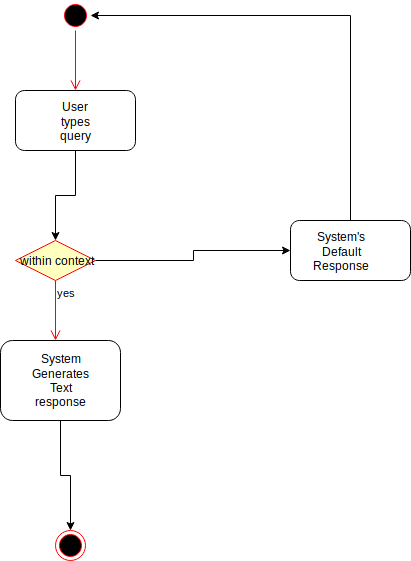
**Figure 4: Car loan Query Processing**

****

**Figure 5:Credit/Debit Cards Query Processing**

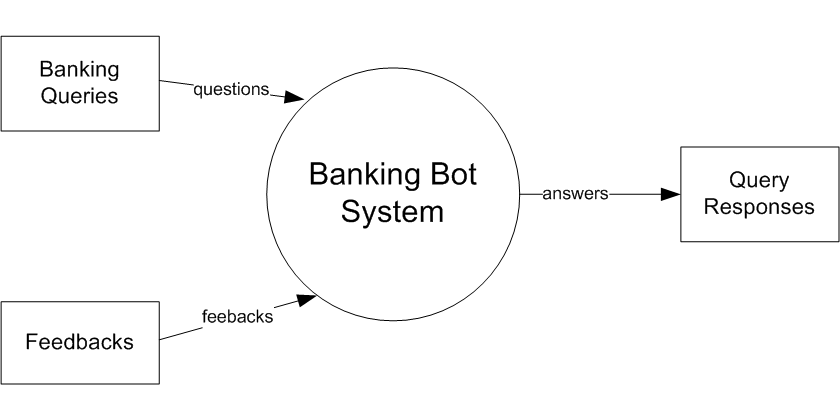
****

**Figure 6: Accounts Query Processing**

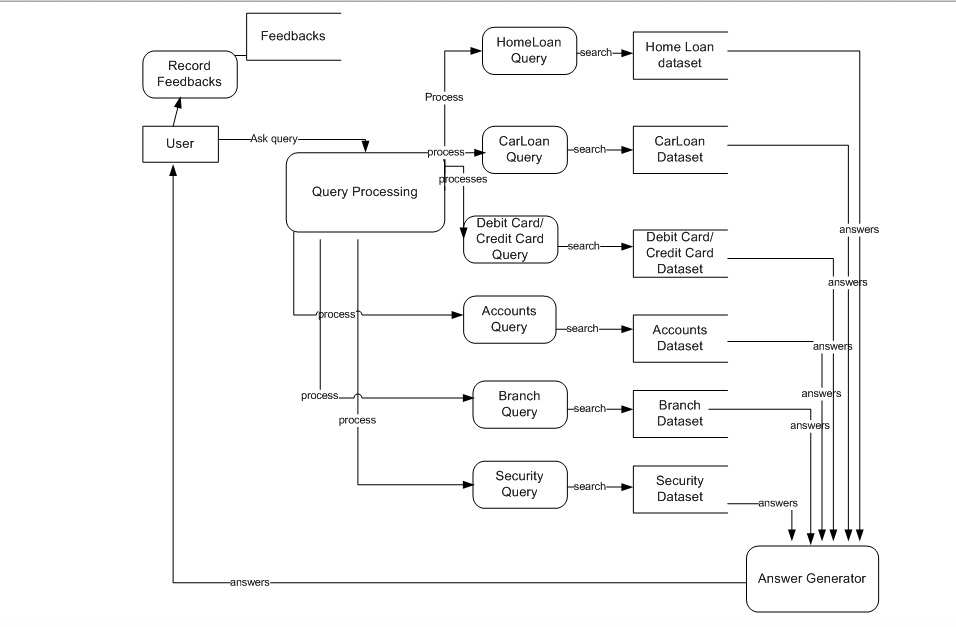
****

**Figure 7: Mutual funds investment**

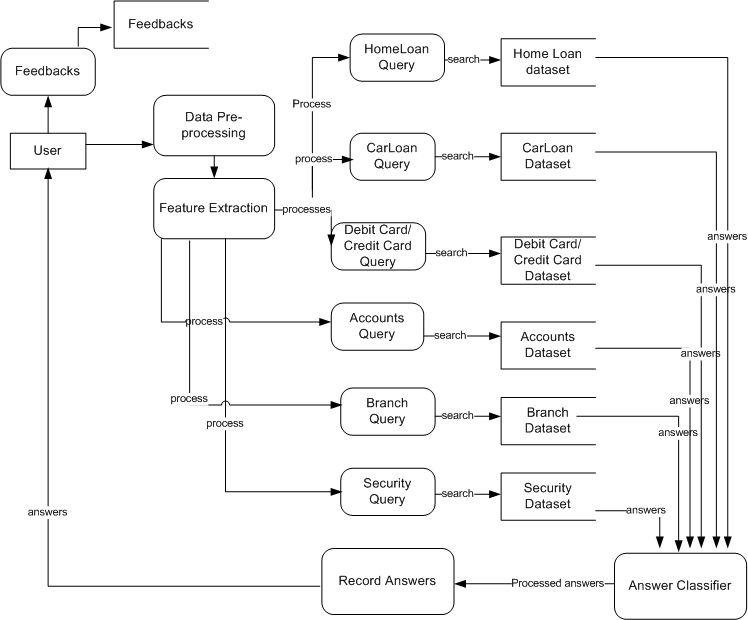
# Design models



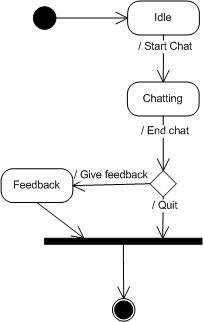
**Figure 8: DFD Level 0**



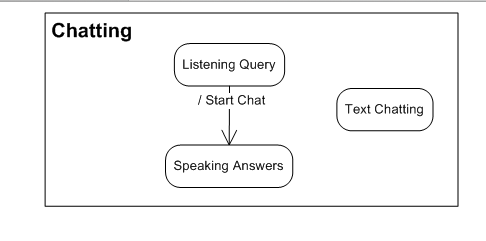
**Figure 9: DFD Level 1**



**Figure 10: DFD Level 2**



**Figure 11: State Transition**



**Figure 12: Sub-State Transition**

# Data design

## Data dictionary

To train the neural network, two files were created where one acted as input and other as output. Input file contained all the questions or the first part of the conversation whereas the output file contained all the respective answers or the other part of the conversation.

# Algorithm & Implementation

For training the model, two files were created where one acted as input and other as output. Input file contained all the questions or the first part of the conversation whereas the output file contained all the respective answers or the other part of the conversation. It consists of two RNN An Encoder and a Decoder. encoder takes a sequence(sentence) as input and processes one symbol(word) at each timestep. Its objective is to convert a sequence of symbols into a fixed size feature vector that encodes only the important information in the sequence while losing the unnecessary information. You can visualize data flow in the encoder along the time axis, as the flow of local information from one end of the sequence to another. Before training, we work on the dataset to convert the variable length sequences into fixed length sequences, by **padding**. We use a few special symbols to fill in the sequence. As the length of the sequence gets larger, we start losing considerable amount of information. This is why the basic seq2seq model doesn’t work well in decoding large sequences. The attention mechanism, introduced in this paper, [Neural Machine Translation by Jointly Learning to Align and Translate](https://arxiv.org/abs/1409.0473), allows the decoder to selectively look at the input sequence while decoding. This takes the pressure off the encoder to encode every useful information from the input.

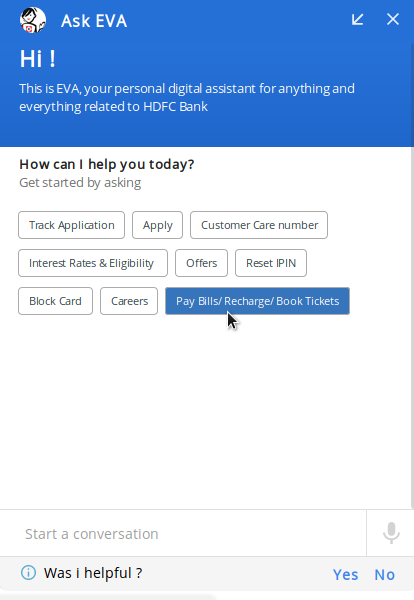
# Software requirements traceability matrix

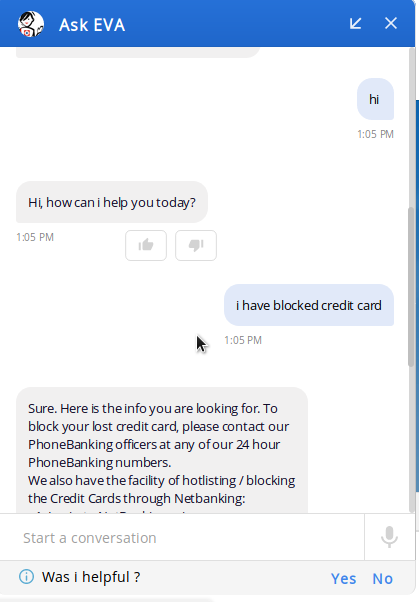
Table 1 Requirements Traceability Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. Number** | **Ref. Item** | **Diagram/Level** | **Functionality** |
| FR01 | DFD | 2/Level 0 | Initiate Chat |
| FR02 | DFD | 3/Level 1 | Text Based Chat |
| FR03-FR04 | DFD | 4/Level 2 | Voice Based Chat |
| FR05 | DFD | 4/Level 2 | Understand Natural Language |
| FR06 | DFD | 3/Level 1 | Search Comparison |
| FR07 | DFD | 4/Level 2 | Correct Response |
| FR08 | DFD | 4/Level 2 | Simplify Input |
| FR09 | DFD | 4/Level 2 | Answer not available |
| FR10 | DFD | 3/Level 1 | Home Loan queries |
| FR11 | DFD | 3/Level 1 | Car Loan queries |
| FR12 | DFD | 3/Level 1 | Credit Card queries |
| FR13 | DFD | 3/Level 1 | Debit Card queries |
| FR14 | DFD | 3/Level 1 | Accounts information |
| FR15 | DFD | 3/Level 1 | Insurance Information |
| FR16 | DFD | 3/Level 1 | Branch Information |
| FR17 | DFD | 3/Level 1 | Security Information |
| FR18 | DFD | 3/Level 1 | Investments procedure |
| FR20 | State Transition | 11 | Quit Chat |
| FR21 | DFD | 4/Level 2 | Feedbacks |
| FR22 | State Transition | 12 | Listening Voice input |

# Human interface design

## Screen images



**Figure 13: Selection**

**Figure 14: Query Handling and feedback**

## 8.2 Screen objects and actions

The user can will write queries and bot will answer these queries. User can give feedback based on the reply given by the user