**Loan Approval Prediction**

Software Development Design (SDD)

Makerere University, College of Computing and Informatics Sciences

Software Engineering

Names:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Student Number | Registration Number | Signature |
| Amoko Ivan | 217014453 | 17/U/2680/EVE |  |
| Senfuka Raymond Wilson | 217012016 | 17/U/9982/PS |  |
| Asiimwe Brenda Angel | 217002455 | 17/U/3014/EVE |  |
| Ssemakula Rajab | 217011090 | 17/U/10165/EVE |  |

Date: 10th July 2019

**TABLE OF CONTENTS**

1. **INTRODUCTION**
   1. Purpose
   2. Scope
   3. Overview
   4. Reference Material
   5. Definitions and Acronyms
2. **SYSTEM OVERVIEW**
3. **SYSTEM ARCHITECTURE**
   1. Architectural Design
   2. Decomposition Description
   3. Design Rationale
4. **DATA DESIGN**
   1. Data Description
   2. Data Dictionary
5. **COMPONENT DESIGN**
6. **HUMAN INTERFACE DESIGN**
   1. Overview of User Interface
   2. Screen Images
   3. Screen Objects and Actions
7. **REQUIREMENTS MATRIX**
8. **APPENDICES**

**List of Figures**

Figure 1: Login form screen

Figure 2: Administrator / Management Dashboard

Figure 3: Loan Applicant’s Dashboard

**List of Tables**

Table 1: Login table

Table 2: Loan Applicant’s Details Table

Table 3: Loan Table

Table 4: Requirements Matrix

1. **INTRODUCTION**
   1. **Purpose**

This software design document describes the architecture and the system design of the Loan Approval Prediction module of the system for predicting loan eligibility of loan applications.

* 1. **Scope**

The system can visualize past loan application data to determine an appropriate formula using the Logistic Regression Model that can be used to predict eligibility of the incoming loan applications.

It is based in RStudio, a free and open source integrated development environment for R programming language installed on a computer with the Shiny packages and other relevant packages.

The goal of this system is to predict loan eligibility based on applicant’s information using the Logistic Regression Model.

Benefits

1. Critically reduce on the time used by personnel in determining loan applicant’s eligibility to a loan.
2. Eliminates favoritism and nepotism amidst loan applicants and the money lenders.
3. Unnecessary delays by loan applicant’s in queues are reduced, since it’s a web-based system, users can apply at their own convenience.
   1. **Overview**

This document has 8 sections that explain its use to the user

1. The Introduction defines the system’s objective and the summary of the system functionality so as to give the reader a good understanding of the system goals.
2. The System Overview which gives a general description of the functionality, context and design of the project.
3. The System Architecture has the detailed view of the different conceptual requirements for setting up the system in terms of hardware and software.
4. The Data Design explains how the information domain of your system is transformed into data structures and alphabetically list the system entities along with their types and descriptions.
5. The Component Design describes how the different components of the system interact to satisfy the user needs.
6. The Human interface describes the functionality of the system, display screenshots showing the interfaces from the user’s perspective.
7. The Requirements matrix provides a cross reference that traces components and data structures to the requirements in your SRS document.
8. Appendices contains the glossary where all the key terms used in this document are defined.
   1. **Reference Material**

IEEE Std 1016 2009 (first version 1998)

1. **SYSTEM OVERVIEW**

This concept and system design included the entire process of establishing a relationship between the loan applicant and management, receiving loan request information, analyzing it and determining whether the request is firm to be approved or not so as to be rejected.

It is meant to be user-friendly for the general public (loan applicants), service providers, and staff.

Project goals included:

* Ability to create loan applicant profiles with permissions from admin, view records of current eligibility, know requirements needed for loan application,
* Ability of an applicant to apply for a loan
* Ability of an applicant to enter loan applicant information
* Ability of the system to receive the applicant loan requests
* Information on all loan requests and ability to schedule loan approvals by management
* Data analysis/monitoring to find efficiencies and influence planning/future implementation in a system-wide feedback loop
* Ability of system to analyze loan requests to determine the ones to approve or reject
* Ability of applicant to notice that the request is either rejected or approved.

1. **SYSTEM ARCHITECTURE**
   1. **Architectural design**

The system uses a client server architecture where a loan applicant through a browser(client) requests for a loan along with relevant information about himself such as credit history, income and so on. The system then sends him a response whether or not he has received the loan(server).

The system uses a 2 tier client server architecture where the client communicated directly to the server with no intervening middleware. In our system a loan applicant talks directly to our system server and is rejected or approved.

1. **DATA DESIGN**
   1. **Data Description**

The system fetches the loan request information from the loan applicant’s form. The loan applicant’s eligibility is determined using the formula that is generated from the logistic regression model on the train and a copy of the loan requests is stored as a list in the MYSQL database.

The rejected loan requests are stored as lists in the database and feedback is sent back to the applicant. The approved loan requests are sent to management and a copy is stored in the database.

Final Management decision on the system approved applications is then sent back to the loan applicant.

* 1. **Data Dictionary**

The data used by the system shall be store in a relational MYSQL based database. Below are the tables and the data values to be stored.

**Login Table**

The table which contains the system user credentials

Table 1

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMN | DATA TYPE | LENGTH | DESCRIPTION |
| username | varchar | 50 | User’s login username |
| password | varchar | 255 | User’s login password |
| User\_type | varchar | 20 | Type of the user either admin or loan applicant |
| Client\_ID | varchar | 30 | The client’s id in the financial institution |

**Loan applicant table**

This table shall store the customer detail provided while filling online application forms.

Table 2

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMN | DATA TYPE | LENGTH | DESCRIPTION |
| Date | date |  | Date of application |
| ApplicantID(PK) | varchar | 20 | Unique applicant ID |
| ApplicantIncome | integer | 30 | Applicant income |
| CoapplicantIncome | integer | 30 | Coapplicant income |
| CreditHistory | integer | 30 | credit history meets guidelines |
| Education | text | 30 | Applicant Education (Graduate/ Under Graduate) |
| Gender | text | 10 | Male/ Female |
| Loan\_ID(FK) | varchar |  | Unique Loan ID |
| LoanAmount | integer | 30 | Loan amount in thousands |
| LoanAmountTerm | integer | 30 | Term of loan in months |
| Marital\_status | text | 15 | Applicant married (Y/N) |
| Number\_of\_dependents | integer | 15 | Number of dependents |
| PropertyArea | text | 30 | Urban/ Semi Urban/ Rural |
| Self\_employed | text | 30 | Self Employed(Y/N) |

**Loan table**

This table shall store the customer status for those eligible for the loan so that the bank can specifically target these customers.

Table 3

|  |  |  |  |
| --- | --- | --- | --- |
| COLUMN | DATA TYPE | LENGTH | DESCRIPTION |
| Date | date |  | Date when the application was submitted |
| Loan\_ID | varchar | 15 | Unique Loan ID |
| Loan\_status | text | 15 | Loan approved (Y/N) |
| LoanAmount | integer | 30 | Loan amount in thousands |
| LoanAmountTerm | integer | 30 | Term of loan in months |
|  |  |  |  |

1. **COMPONENT DESIGN**

**Module 1: User Verification module**

Users of the system must have an account with username and password stored for them to access the functionalities of the system module.

User credentials have to be verified in order to fully access the system.

Module Inputs:

Username

Password

**Psuedocode**

Load the system

Enter username and password

If username or password not in system:

Verification failed

Retry login

Else:

Verification successful

Loan Applicant

Login

Enter details

For each entry if not null:

Proceed to submit

Else:

Enter the details

Management

Login

Views system approved loans

Accepts or rejects depending on institutions standard

Sends feedback

**Module 2: Data Cleaning**

The loan applicant’s information entered must be validated so as to eliminate noise in the information which eliminated errors in the predictions by the system. The spaces in the column names must be replaced with default values.

Module Inputs:

ApplicantIncome, CoapplicantIncome, Education

Gender among others listed in the Data dictionary section 4

**Module 3: Graphical Representation of results**

The analytical will be presented in a graphical view to the. A number of graphical features including but not limited to histograms, line, bar graphs for the past loan applications.

graphs, pie charts, box plots will be used by this module to display data to the user. The user

will select the tool they wish to use to view the results.

1. **HUMAN INTERFACE DESIGN**
   1. **Overview of User Interface**

Interface Architecture

This system will interface with the Database data to validate the user credentials so as to ensure that only authorized users are accessing the system. When the user has entered his/her login credentials, the details will be validated and user can use the system.

System users will be required to have accounts which will be added to the system by the system administrator. On attempt to access the system, the user will be by entering his/her credentials.

The loan Applicant on logging in, will be presented with the user dashboard, from which he/ she would have to fill in a loan application form.

The Administrator of the financial institution is also presented with a specific dashboard through which he can manage the operations of the system module.

* 1. **Screen Images**

**Login Screen image**

Login interface through which users can access the different components of the module on validation

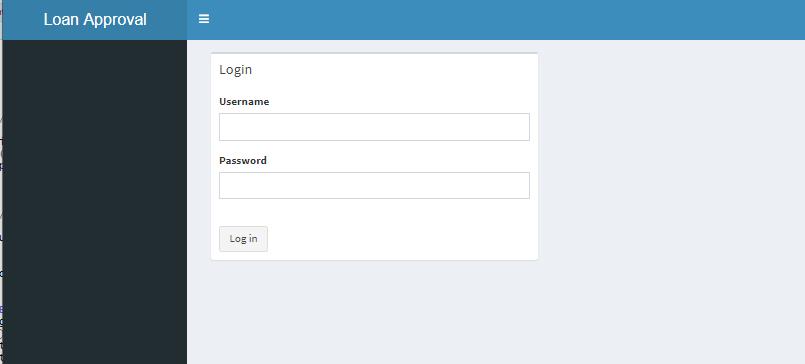


Figure 1

**Administrator’s Dashboard**

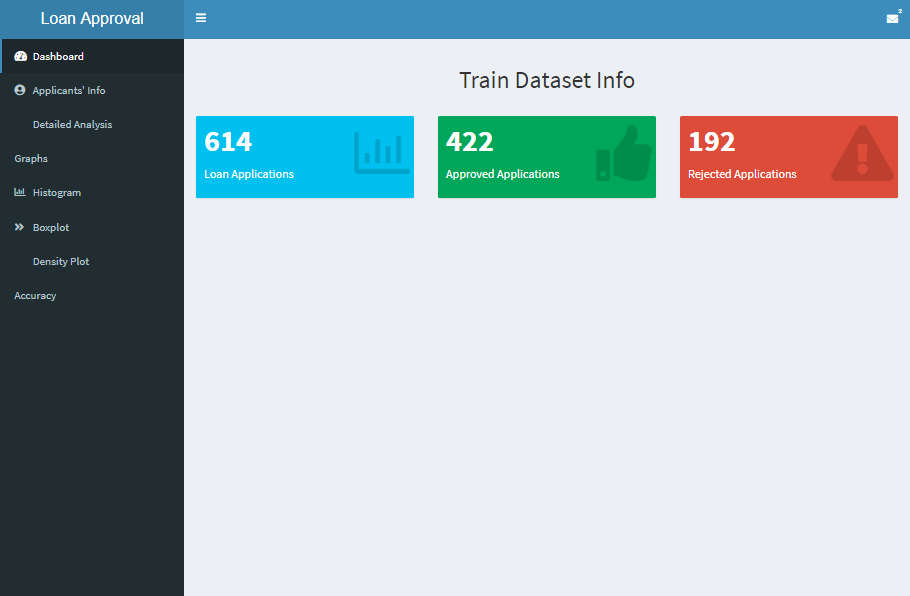


Figure 2

**Loan Applicant’s Dashboard**

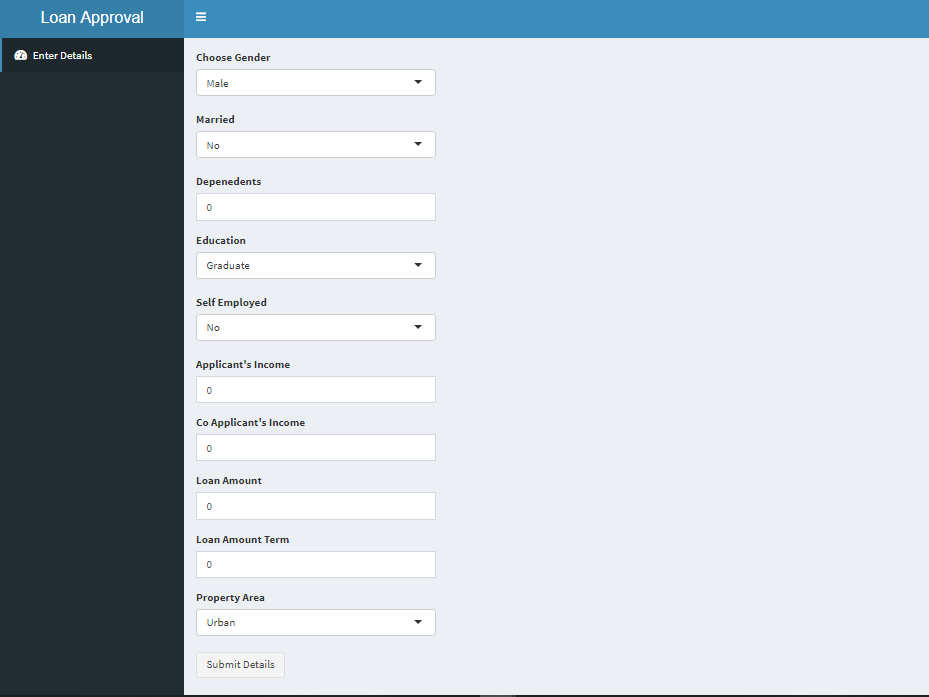


Figure 3

* 1. **Screen Objects and Actions**

**Login Screen image**

Login interface provides as interface through which users can log into the system and can access the different components of the module on validation.

**Administrator’s Dashboard**

Dashboard presented to the Administrator on login validation.

**Loan Applicant’s Dashboard**

The Loan applicants on validation, are redirected to the user dashboard in which they are able to fill in the loan application form.

1. **REQUIREMENTS MATRIX**

Table 4

|  |  |
| --- | --- |
| FUNCTIONALITY REQUIREMENTS | REFERENCE SECTION IN SRS |
| User Logs into the system | 4.1.1 System Feature ( Login ) |
| Processing of loan applicant’s details | 4.1.2 System Feature (Analysis and processing) |
| Results Retrieval | 4.1.3 System Feature ( View Analyzed data) |

1. **APPENDICES**
   1. Glossary
2. Graph: A system of nodes connected in pairs by edges. Often sub-divided into directed graphs or undirected graphs according to whether the edges have an orientation or not. Mixed graphs include both types of edges.
3. Node: A node is (together with edges) one of the two basic units out of which graphs are constructed. Nodes of graphs are often considered to be atomic objects, with no internal structure.
4. Edge: An edge is (together with vertices) one of the two basic units out of which graphs are constructed. Each edge has two (or in hyper graphs, more) vertices to which it is attached, called its endpoints.