B.K. Jewellers

**High Level Design (HLD)**

**Document for**

**“Accounting” part.**

Document NO: 001

Date: 11-02-2020

Prepared by:

Group 2

* **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision Number** | **Revision Date** | **Summary of changes** | **Author** |
| **1** | 11/02/2020 | Document made. | Ghanshyam gamit |
| **2** | 18/04/2020 | * **Table of content:** Page numbers added. * **(1.3):** Software interface deleted from list. | Rudra soni |
| **3** |  |  |  |
|  |  |  |  |

* **Table of Contents** -

|  |  |  |
| --- | --- | --- |
| Sr no. | Content | Page no. |
| 1. | Introduction |  |
|  | * 1. Objective | 4 |
|  | 1.2 Scope | 4 |
|  | 1.3 Overview | 4 |
|  |  |  |
| 2. | General description |  |
|  | 2.1 Product perspective | 4 |
|  | 2.2 Tools used | 5 |
|  |  |  |
| 3. | General constrains | 5 |
|  |  |  |
| 4. | Assumptions |  |
|  | 4.1 Peripheral assumption | 5 |
|  | 4.2 Result assumption | 6 |
|  | 4.3 Knowledge assumption | 6 |
|  |  |  |
| 5. | Design details |  |
|  | 5.1 Application architecture | 6 |
|  | 5.2 Screen presentation | 7 |
|  | 5.3 Standards | 7 |
|  | 5.4 User interface | 7 |
|  | 5.5 Security | 7 |
|  | 5.6 Resource utilization | 8 |
|  | 5.7 Help | 8 |

1. **Introduction**
   1. Objective:

The purpose of this High Level Design (HLD) document is to add required details to the current project to make a suitable model for coding. This document can also be used as reference manual for how modules work at high level.

* 1. Scope:

The HLD document defines the full architecture of the “accounting” department in software.

* 1. Overview:

The HLD document will describe following:

* All the design aspects and defined in detail.
* User interface.
* Resource utilization.
* Design features & architecture of project.

1. **General description**
   1. Product perspective:

The working of “accounting” department is build with several components some of them are programmed and others are implemented from open-source programs.

This section will allow only one user to work with software on one machine.

There are two users, first is “accountant” which can view

* billing,
* pending payments,
* Account maintenance table.
* Other tax maintenance.

The second type of user is “account manager” who has the ability to view daily transactions, any suggestion and reporting to higher authority.

* 1. Tools used:
* Unified modelling language (UML) design programs to generate all diagram.
* Backend data-base is NoSQL type based.

1. **General constrains**

The “account” department must be user friendly and partially automatic.

Accountant should not be required to know working of any other departments within the organization.

Once any entry is saved accountant do not make any changes to it, mistakes can only be edited by account- manager only.

1. **Assumptions**
   1. Peripheral assumption:

The “account” department can work on only Microsoft Windows (7/8.1/10/above) . The required specifications are at least 4 GB of RAM and 100 GB of free storage space.

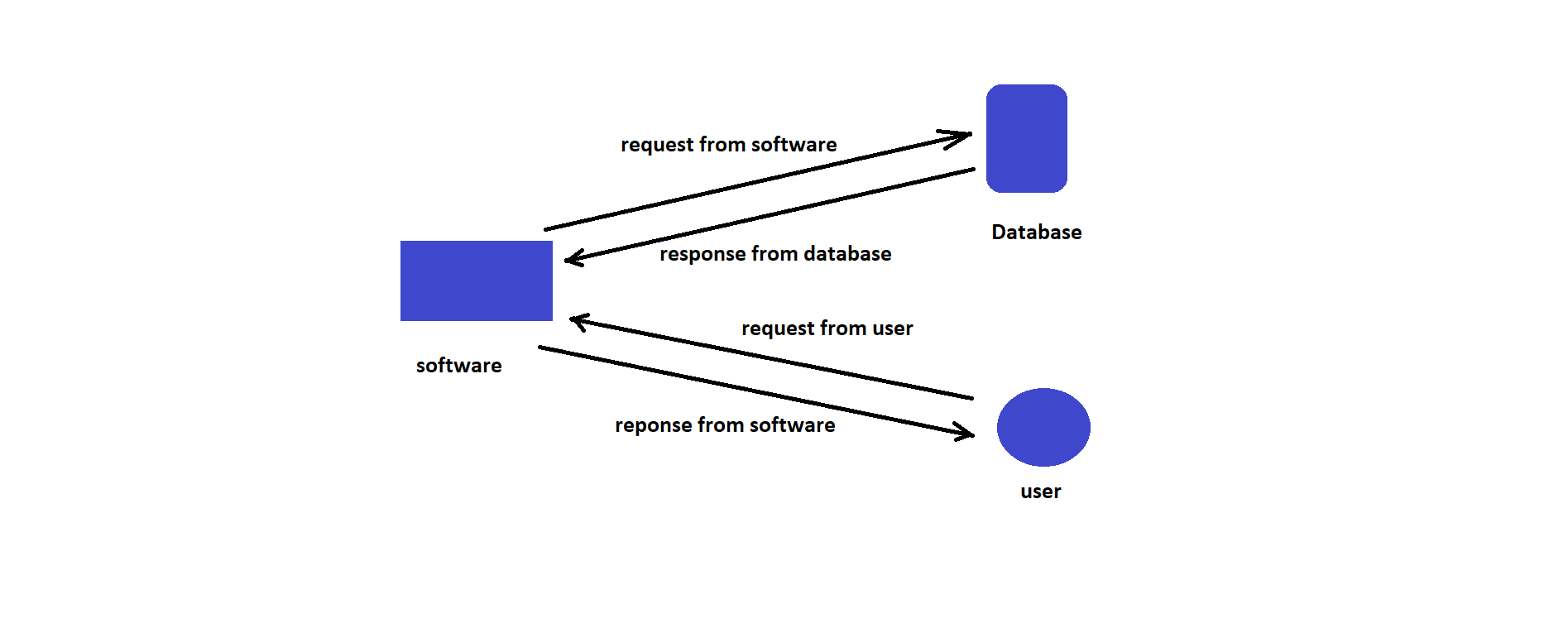
* 1. Result assumption:

The goal is to make the working within organization will follow the strict rules and procedures with less efforts and make the process simpler as much as possible.

* 1. Knowledge assumption:

The final assumption is that the person is operating the software is aware of basic usage of computers and at least 80% awareness of rules and procedures of the organization as well as 95% of knowledge in their working expertises.

1. **Design details** 
   1. Application architecture:



* 1. Screen presentation:

On “accountant” side information will include

* saved bills,
* pending payments,
* GST rates,

On “account manager” side information will include

* Completed invoice,
* pending invoices,
* Bank account maintenance.
* Salary status.
  1. Standards:
* Inputs:- through text fields and stored in database.
* Security:- username and password are required.
* Quality:- by keeping simple and direct interface quality should be kept at a maximum.
* Accuracy:- the accuracy standard should always keep at very high when performing any operation.
  1. User interface:

The user interface is very simple plain layout with little to no graphics. It will display information very clearly for the users.

* 1. Security:

A username and password will be mandatory to log into the system as well as the software.

The completed invoice numbers shouldn’t be shown in any invoice afterwards.

* 1. Resource utilization:

When any task is performed, it will likely to use recommended processing power until that task get finished.

* 1. Help:

Help will come in the form of all documentation created prior to coding, which explain the intended user. Detailed instructions will be written in it.