**~ Change table ~**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. no. | Change in colour | Description | Suggestion | Suggested to | Remarks |
| 1. |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

B.K. Jewellers

**High Level Design (HLD)**

**Document for**

**“Sales management” part.**

Document NO: 001

Date: 09-02-2020

Prepared by:

Group 2

* **Table of Contents** -

|  |  |
| --- | --- |
| 1. | Introduction |
|  | * 1. Objective |
|  | 1.2 Scope |
|  | 1.3 Overview |
|  |  |
| 2. | General description |
|  | 2.1 Product perspective |
|  | 2.2 Tools used |
|  |  |
| 3. | General constrains |
|  |  |
| 4. | Assumptions |
|  | 4.1 Peripheral assumption |
|  | 4.2 Result assumption |
|  | 4.3 Knowledge assumption |
|  |  |
| 5. | Design details |
|  | 5.1 Application architecture |
|  | 5.2 Screen presentation |
|  | 5.3 Standards |
|  | 5.4 User |
|  | 5.5 Security |
|  | 5.6 Resource utilization |
|  | 5.7 Help |

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 “Sales” department in software.

* 1. Overview:

The HLD document will describe following:

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

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

The working of “Sales” 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 system. It will also set up by using two users, first is “Salesman” which view only transfers,

stock- in, stock-out, billing and sales history of current day, the second type of user is “Sales 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 “Sales” department must be user friendly and partially automatic.
* Salesman should not be required to know working of any other departments within the organization.
* Once any entry is saved salesman do not make any changes to it, mistakes can only be edited by sales- manager only.

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

The “Sales” department can work on only Microsoft Windows (7/8.1/10/above) and LINUX (12.0 LTS/above) systems. 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.

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

**---------(N/A)---------**

Note: insufficient knowledge.

* 1. Screen presentation:

On “salesman” side information will include

* saved bills,
* stock-in,
* stock-out,
* item transfer information.

On “Sales manager” side information will include

* history of transferred items,
* sold items,
* item statuses,
* pending events information.
  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.
  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 sold items names/codes shouldn’t be shown in any invoice afterwards.

* 1. Resource utilization:

When any task is performed, it will likely to use all the processing power available 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.