GUESTHOUSE MANAGEMENT

SYSTEM

PROJECT REPORT

BY AMISHAPRIYA SINGH

2016

PROJECT REPORT

ON

**GUESTHOUSE MANAGEMENT SYSTEM**

IN ABAP

(ADVANCED BUSINESS APPLICATION PROGRAMMING)



SESSION 2016-17

BY:

AMISHAPRIYA SINGH

B. Tech (CSE), 2nd Year

**Under the Guidance of**

Mr. Ajit Singh

DGM (Programming)

I/C Programming Team,

Project ICE, ONGC

**CERTIFICATE**

This is to certify that Amishapriya Singh, a student of Indira Gandhi Delhi Technical University for Women, New Delhi has undergone project training in our organization for 4 weeks starting from 1st. July 2016 to 29th July 2016.

She has successfully completed the training on SAP ABAP, under the supervision of Mr. Ajit Singh, DGM (Programming) and under the guidance of Mr. Mahesh Kumar Pareek, Manager (Programming).

During the training, she has worked on ABAP/4 a software development module on the SAP ECC 6.0 ERP platform and covered various topics related to SAP R/3 Systems. She has successfully completed a project on “GUESTHOUSE MANAGEMENT SYSTEM”.

We have observed that her work has been excellent and appreciate her sincere learning. She has performed with energy and enthusiasm.

We wish her all the best for her future endeavours.

Mr. Ajit Singh

DGM (Programming)

I/C Programming Team,

Project ICE, ONGC

**ACKNOWLEDGEMENT**

I take this opportunity to express my profound sense of gratitude and appreciation to all those who helped me throughout the duration of this project.

First and foremost, I would like to thank **Mr. Dharam Raj, GM – Head Project ICE**, for encouragement, support and providing necessary facilities.

I would also like to express my thanks to **Mr. Ajit Singh, DGM (Programming) I/C – ABAP Team, Project ICE** for providing expert supervision for this project and **Mr. Mahesh Kumar Pareek, Manager (Programming)** for the guidance.

I am truly thankful to the entire Programming team in Project ICE for their support and timely help in solving problems related to ABAP/4 language.

# **Amishapriya Singh**

**LIST OF CONTENTS**

|  |  |
| --- | --- |
| **CONTENTS** | **PAGE NUMBER** |
| Abstract | 6 |
| About ONGC | 7 |
| Project ICE | 10 |
| About SAP | 11 |
| SAP R/3 used in ONGC | 12 |
| Project Overview | 16 |
| Hardware and Software Requirements | 17 |
| Program Flow | 18 |
| Design and Working | 19 |
| Project Screenshots | 20 |
| Validation | 25 |
| Table Overview | 26 |
| Program Code | 30 |
| Future Scope | 46 |
| References | 47 |

**ABSTRACT**

**Guest house Management** is about keeping the records of all the guesthouses of ONGC across India and allows the employees to make booking based on their preference of location, room type and dates, and the administrator to keep a check on all the bookings made using a report.

The employee enters his/her preferences for the room and checks the availability status of the room. If the room is unavailable, he/she can change the dates. If the room is available, the employee is asked to verify his/her identity and the room is booked under his/her unique personnel number.

The Admin who oversees the bookings can generate report as per his/her choice and view all the bookings made in the selection category.

**Oil and Natural Gas Corporation and Project ICE**

**Overview of the Company**

Oil and Natural Gas Corporation Limited (ONGC) is an Indian multinational oil and gas company headquartered in Dehradun, Uttarakhand, India. It is a Public Sector Undertaking (PSU) of the Government of India, under the administrative control of the Ministry of Petroleum and Natural Gas. It produces around 69% of India’s crude oil (equivalent to around 30% of the country’s total demand) and around 62% of the natural gas.

ONGC was founded on 14 August 1956 by Government of India, which currently holds a 68.94% equity stake. It is involved in exploring for and exploiting hydrocarbons in 26 sedimentary basins of India, and owns and operates over 11,000 kilometres of pipelines in the country. Its international subsidiary ONGC Videsh currently has projects in 17 countries. ONGC has discovered 6 of the 7 commercially producing Indian Basins, in the last 50 years, adding over 7.1 billion tonnes of In-place Oil & Gas volume of hydrocarbons in Indian basins. Against a global decline of production from matured fields, ONGC has maintained

production from its brown fields like Mumbai High, with the help of aggressive investments in various IOR (Improved Oil Recovery) and EOR (Enhanced Oil Recovery) schemes. ONGC has many matured fields with a current recovery factor of 25-33%.

**History of the Company**

In 1955, Government of India decided to develop the oil and natural gas resources in the various regions of the country as part of Public Sector development. With this objective, an Oil and Natural Gas Directorate was set up in 1955 under the then Ministry of Natural resources and Scientific Research. The department was constituted with a nucleus of geoscientists from the Geological survey of India.

Soon, after the formation of the Oil and Natural Gas Directorate, it became apparent that it would not be possible for the Directorate with limited financial and administrative powers to function efficiently. So in August, 1956, the Directorate was raised to the status of a commission

with enhanced powers, although it continued to be under the

government. In October 1956, the Commission was converted into a statutory body by an act of Parliament, which enhanced powers of the commission further. The main functions of the Oil and Natural Gas Commission subject to the provisions of the Act, were “to plan, promote, organize and implement programmes for development of Petroleum Resources and the production and sale of petroleum and petroleum Government may, from time to time, assign to it”. The act further outlined the activities and steps to be taken by ONGC in fulfilling its mandate.

Since its inception, ONGC has been instrumental in transforming the country’s limited upstream sector into a large viable playing field, with its activated spread throughout India and significantly in overseas territories. In the inland areas, ONGC not only found new resources in Assam but also establishes new oil province in Cambay basin (Gujarat), while adding new petroliferous areas in the Assam-Arakan Fold Belt and East coast basins (both inland and offshore).

ONGC went offshore in early 70's and discovered a giant oil field in the form of Bombay High, now known as Mumbai High. This discovery, along with subsequent discoveries of huge oil and gas fields in Western offshore changed the oil scenario of the country. Subsequently, over 5 billion tonnes of hydrocarbons, which were present in the country, were discovered. The most important contribution of ONGC, however, is its self-reliance and development of core competence in E&P activities at a globally competitive level.

The liberalized economic policy, adopted by the Government of India in July 1991, sought to deregulate and de-license the core sectors (including petroleum sector) with partial disinvestments of government equity in Public Sector Undertakings and other measures. As a consequence, thereof, ONGC was re-organized as a limited Company under the Company's Act, 1956 in February 1994.

After the conversion of business of the erstwhile Oil & Natural Gas Commission to that of Oil & Natural Gas Corporation Limited in 1993, the Government disinvested 2 per cent of its shares through competitive bidding. Subsequently, ONGC expanded its equity by another 2 per cent by offering shares to its employees.

**Project ICE**

Project ICE (Information Consolidation for Efficiency) has been initiated to implement ERP in the following major areas:

Enterprise Management: Strategic enterprise management, Business Intelligence, Managerial/Accounting, Joint Venture Management.

Exploration and Production: Exploration and Appraisal, Development and Production.

Supply: Supply Chain Optimization.

Manufacturing: Gas and Fuel Manufacturing.

Business Support: HR core functions, HR – Analytics, Procurement, Treasury/Corporate Finance Management, Fixed Asset Management, E&C and Maintenance.

Around 140 officers have been identified as Business Analysts, Business Configuration and ABAP Developers.

**ABOUT SAP**

**Overview**

SAP AG was founded in 1972 by five German engineers with IBM in Mannheim, Germany; interestingly four of the founders-- Hasso Plattner, Dietmar Hopp, Klaus Tschira, and Hans Werner Hector -- were still with SAP in early 1996. When an IBM client asked IBM to provide enterprise wide software to run on its mainframe, the five began writing the program only to be told the assignment was being transferred to another unit. Rather than abandon the project altogether, they left IBM and founded SAP in Walldorf, near Heidelberg. While the company originally took its name from the abbreviation for *Systemanalyse und* *Programmenentwicklung* (systems analysis and program development), SAP eventually came to stand for *Systeme, Anwendungen, und* *Produkte in Datenverarbeitung* (systems, applications, and products in data processing).

The original SAP idea was to provide customers with the ability to interact with a common corporate database for a comprehensive range of applications. Gradually, the applications have been assembled and today many corporations, including IBM and Microsoft, are using SAP products to run their own businesses.

It operates in three geographic regions u2013 EMEA, which represents Europe, Middle East and Africa; the Americas (SAP America, headquartered in Newtown Square, Pennsylvania), which represents both North America and Latin America; and Asia Pacific Japan (APJ), which represents Japan, Australia, India and parts of Asia. In addition, SAP operates a network of 115 subsidiaries, and has R&D facilities around the globe in Germany, North America, Canada, China, Hungary, India, Israel and Bulgaria.

SAP systems provide the basis for building applications that implement business processes. The SAP applications are not dependent on the type of operating system and databases to a great extent. Applications can be built in either ABAP or JAVA.

**SAP R/3 USED IN ONGC**

SAP R/3 is the former name of the enterprise resource

planning software produced by the German corporation SAP AG (now SAP SE). It is an enterprise-wide information system designed to coordinate all the resources, information, and activities needed to complete business processes such as order fulfilment, billing, human resource management, and production planning.

The first version of SAP's flagship enterprise software was a financial Accounting system named R/1 called as YSR. This was replaced by R/2 at the end of the 1970s. SAP R/2 was in a mainframe based business application software suite that was very successful in the 1980s and early 1990s. It was particularly popular with large multinational European companies who required soft-real-time business applications, with multicurrency and multi-language capabilities built in. With the advent of distributed client–server computing SAP AG brought out a client–server version of the software called SAP R/3 (The "R" was for "Real-time data processing" and 3 was for 3-tier). This new architecture is compatible with multiple platforms and operating systems, such as Microsoft Windows or UNIX. This opened up SAP to a whole new customer base.

Sap based the architecture of R/3 on a three tier client/server model:-



**1) Presentation Server**

The presentation server is actually a program named “sapgui.exe”. It is usually installed on a user’s workstation. To start it, the user double-clicks on an icon on the desktop or chooses a menu path. When started, the presentation server displays the R/3 menus within a window. This window is commonly known as the SAPGUI, or the user interface (or simply, the interface). The interface accepts input from the user in the form of keystrokes, mouse-clicks, and function keys, and sends these requests to the application server to be processed.

**2) Application Server**

An application server is a set of executables that collectively interpret the ABAP/4 programs and manage the input and output for them. When an application server is started, these executables all start at the same time. When an application server is stopped, they all shut down together. The number of processes that start up when you bring up the application server is defined in a single configuration file called the application server profile. The application server exists to interpret ABAP/4 programs, and they only run there-the programs do not run on the presentation server. An ABAP/4 program can start an executable on the presentation server, but an ABAP/4 program cannot execute there.

3) **Database server**

The database server handles the user’s request for addition, retrieval and modifications in the data. It is also responsible for maintaining concurrency control.

Project overview

Product perspective

The product will be able to manage tenders in a user friendly manner. This system will manage the tenders at various levels. The tenders can be created, updated, displayed and released.

**Project Overview**

The product will be able to manage bookings and record management of ONGC guesthouses across India. The employees can login to the Guesthouse Management System and book rooms using their unique personnel number. Their access to the central database is restricted to their information only.

This project includes:

 **BOOKING**

The user can check the availability of room according to his/her choice of location, room type and dates. If the room is available, they can confirm booking by verifying themselves as employees of ONGC.

**REPORT GENERATION**

The admin can generate reports to view bookings. The data displayed is subject to the admin preferences. A selection-screen is provided for the admin to enter the criteria that the displayed data must satisfy.

**HARDWARE REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **HARDWARE COMPONENTS** | **MINIMUM REQUIREMENT** |
| 1. | MOTHERBOARD | 965-GV CHIPSET |
| 2. | HARD DISK | 2 GB |
| 3. | RAM | 512 |

**SYSTEM/SOFTWARE REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **SOFTWARE COMPONENTS** | **MINIMUM REQUIREMENT** |
| 1. | OPERATING SYSTEM | HP UNIX SERVER/WINDOW FOR CLIENT |
| 2. | APPLICATION | SAP-ECC 6.0 |
| 3. | ENVIRONMENT | ABAP |

**Program Flow**

SAP LOGIN: USER AUTHENTICATION

RUN T-CODE: ZAMGH

GUESTHOUSE

MANAGEMENT SYSTEM

LOG IN

ADMIN

EMPLOYEE

Generate report based on personnel number, dates, guesthouse.

View all allotments.

Check room availability based on Location, dates and room type.

Book room using personnel number.

**DESIGN AND WORKING**

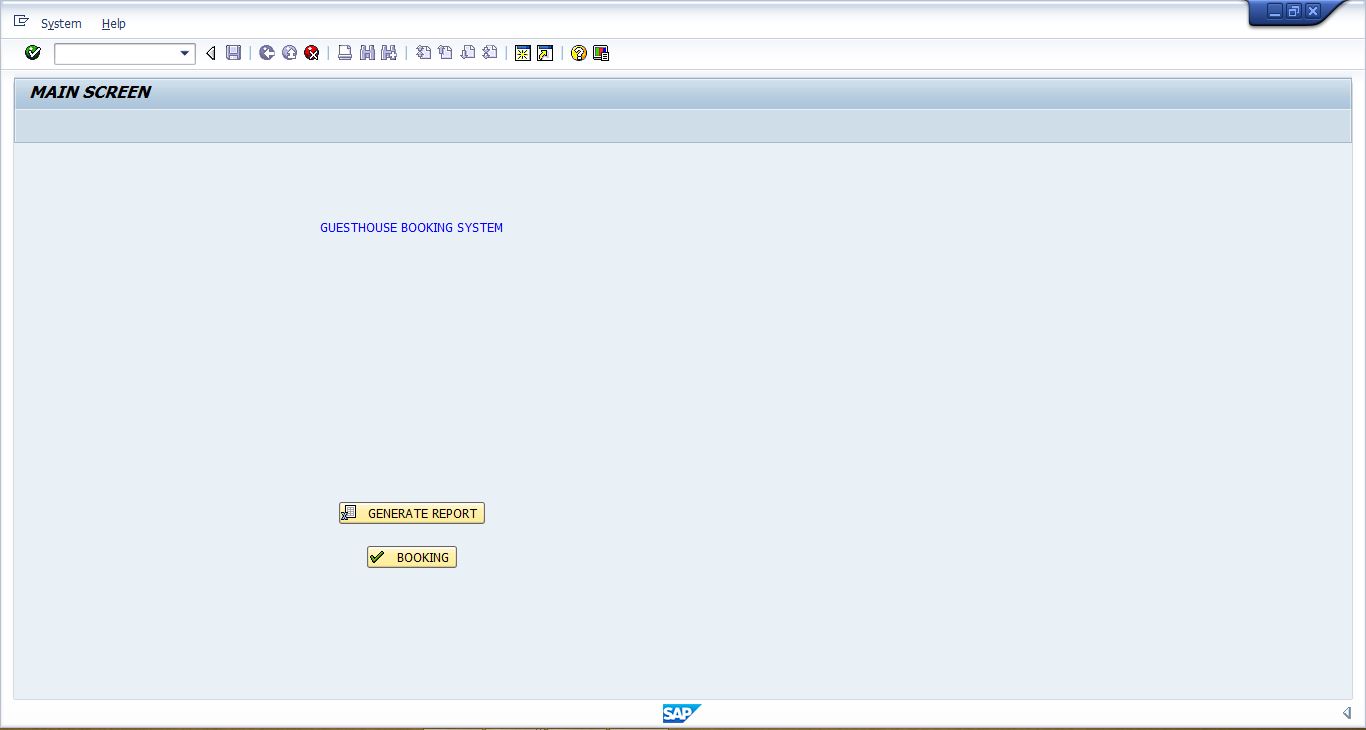
ABAP WORKING

All ABAP programs reside inside the SAP database. They are not stored in separate external files like Java or C++ programs. In the database all ABAP code exists in two forms: source code, which can be viewed and edited with the ABAP Workbench tools, and generated code, a binary representation somewhat comparable with Java byte code. ABAP programs execute under the control; of the runtime system, which is a part of the SAP kernel. The runtime system is responsible for processing ABAP statements, controlling the flow logic of screens and responsible for processing events (such as a user clicking on a screen button); in this respect it can be seen as a Virtual Machine comparable with JAVA VM. A key component of the ABAP runtime system is the Database Interface, which turns database-independent ABAP statements ("Open SQL") into statements understood by the underlying DBMS ("Native SQL"). The database interface handles all the communication with the relational database on behalf of ABAP programs; it also contains extra features such as buffering of tables and frequently accessed data in the local memory of the application server.

**PROGRAM SCREENSHOTS**

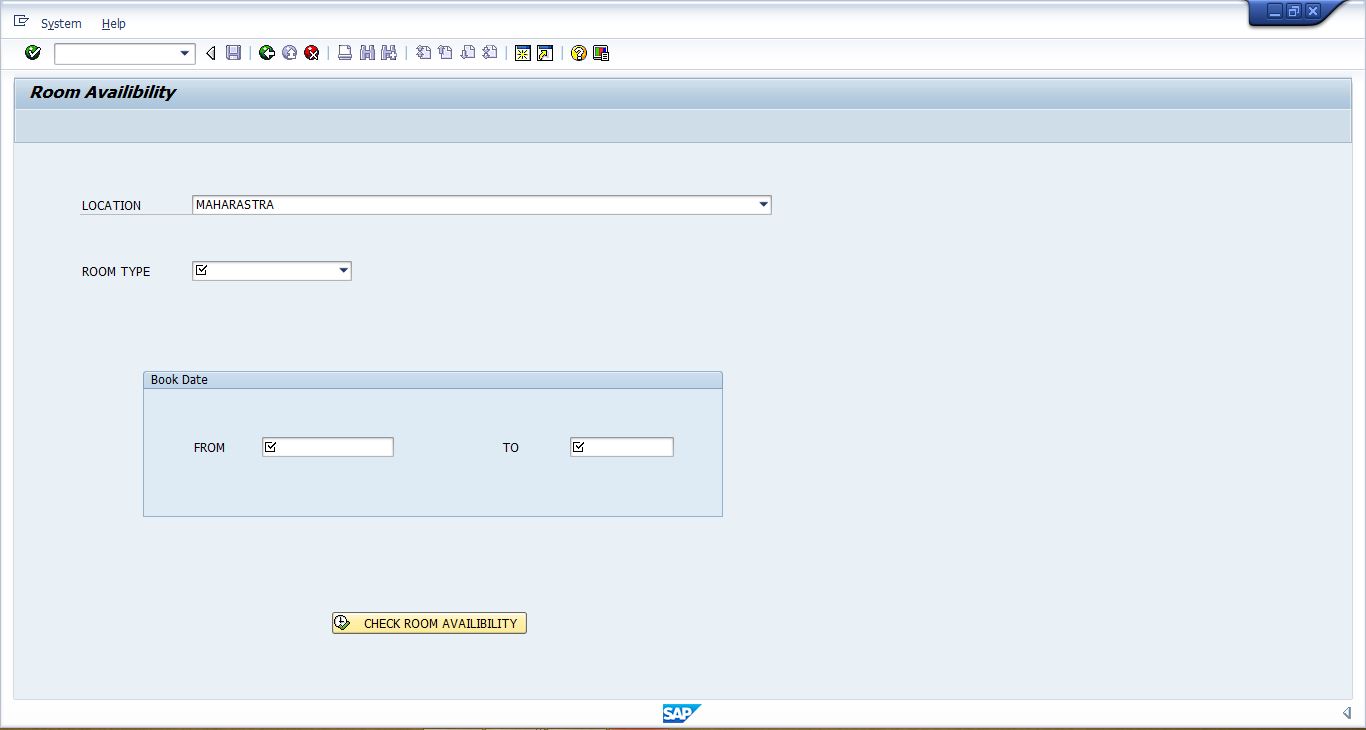
**Main Screen:**

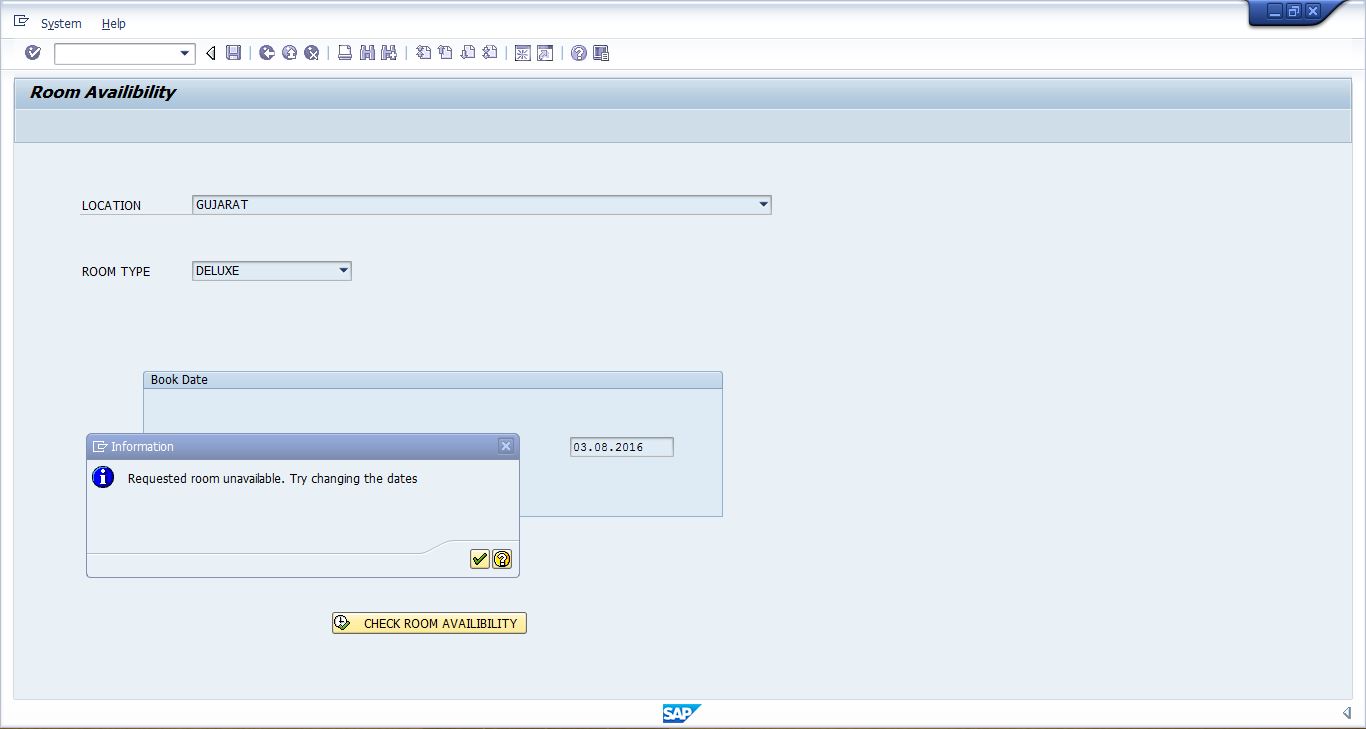
Generate a report or book a room.

****

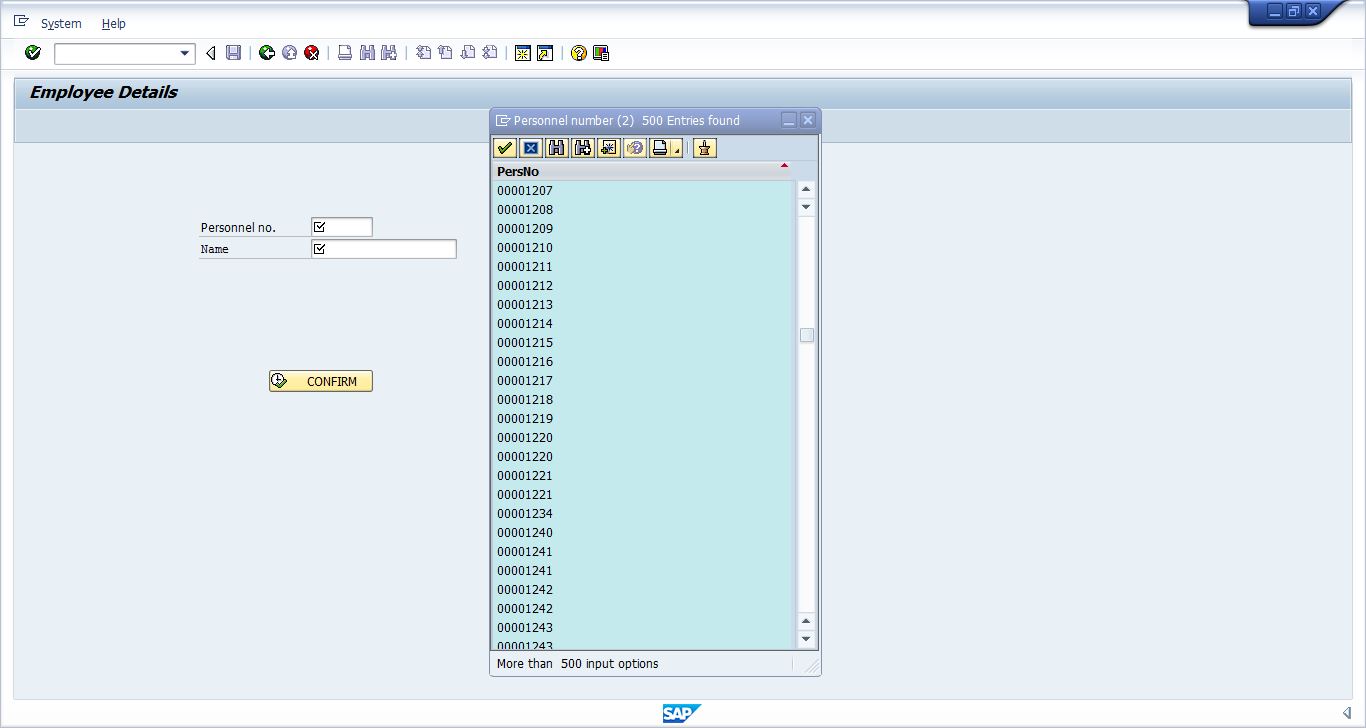
**BOOKING:**

Enter your preferred location, room type, and dates to check the availability of room.

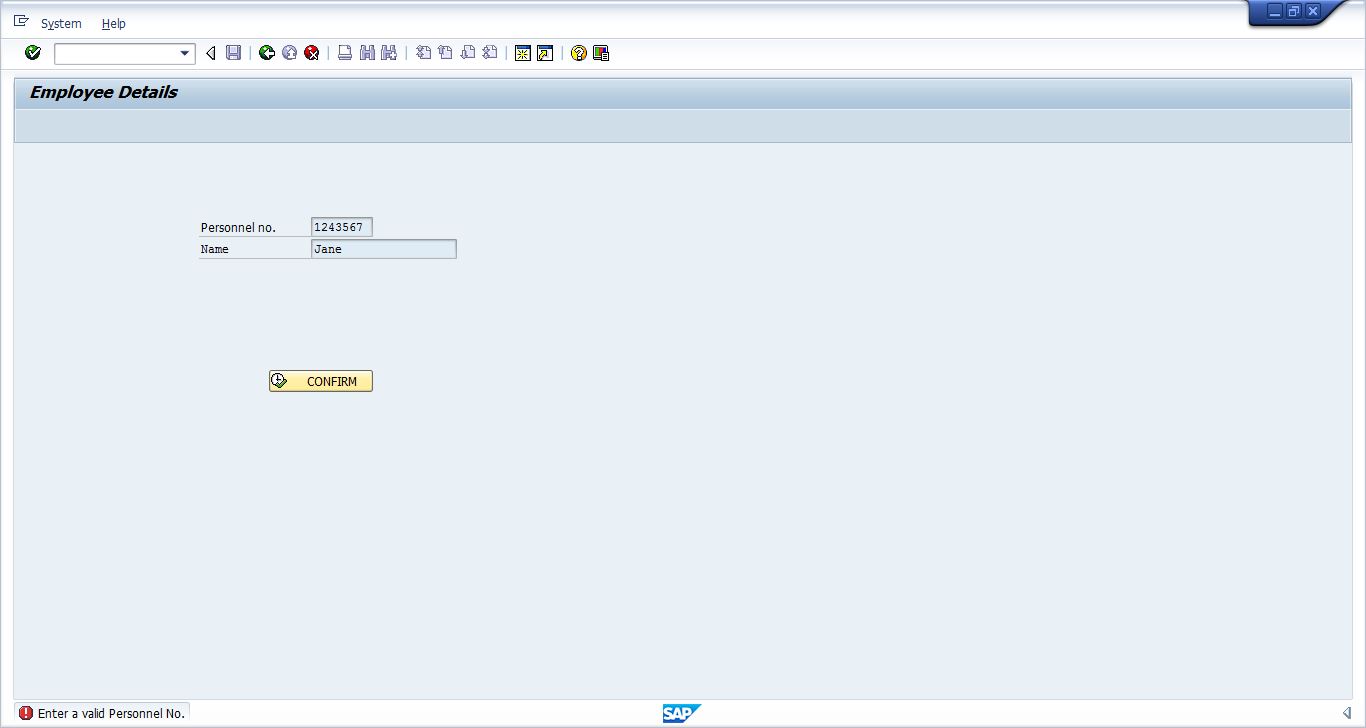
All fields are required. Location and Room Type are drop downs.

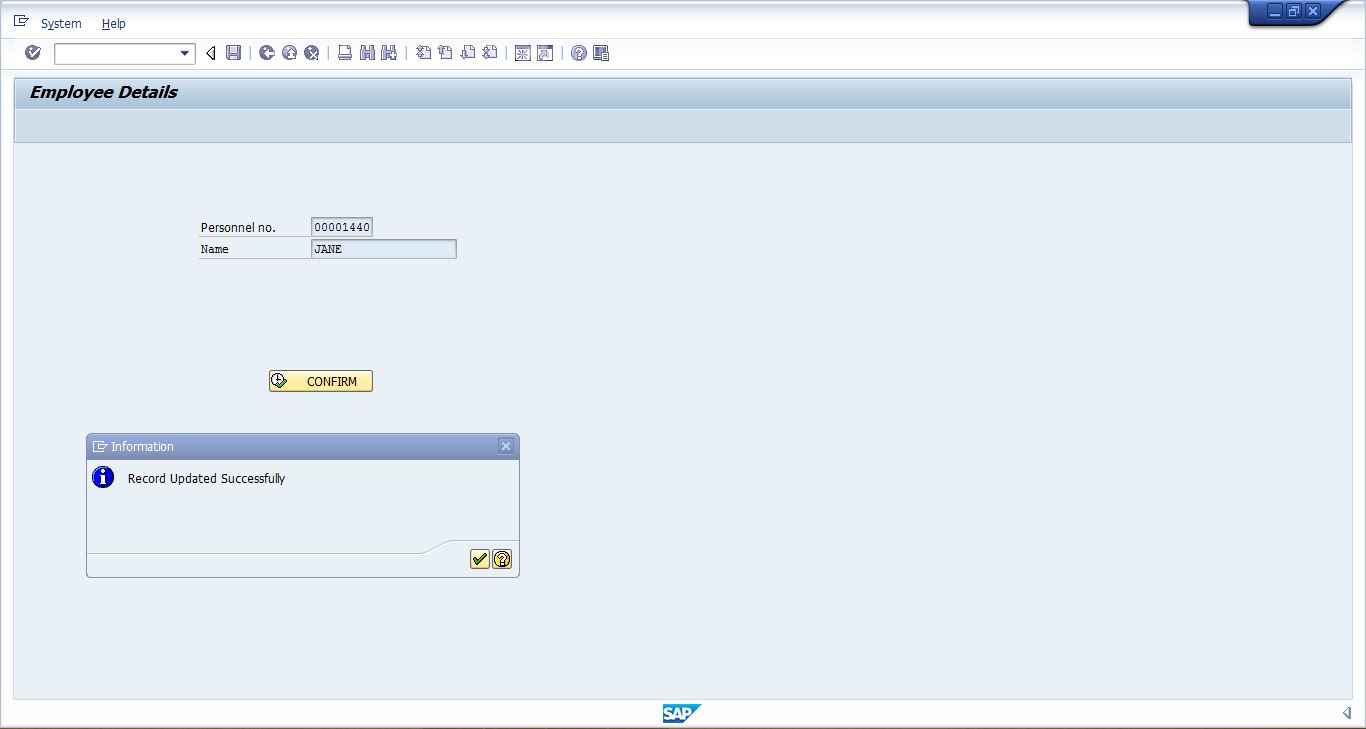
If the room is not available, the user is prompted to change the dates.

If the room is available, the user is taken to the detains screen to confirm booking.



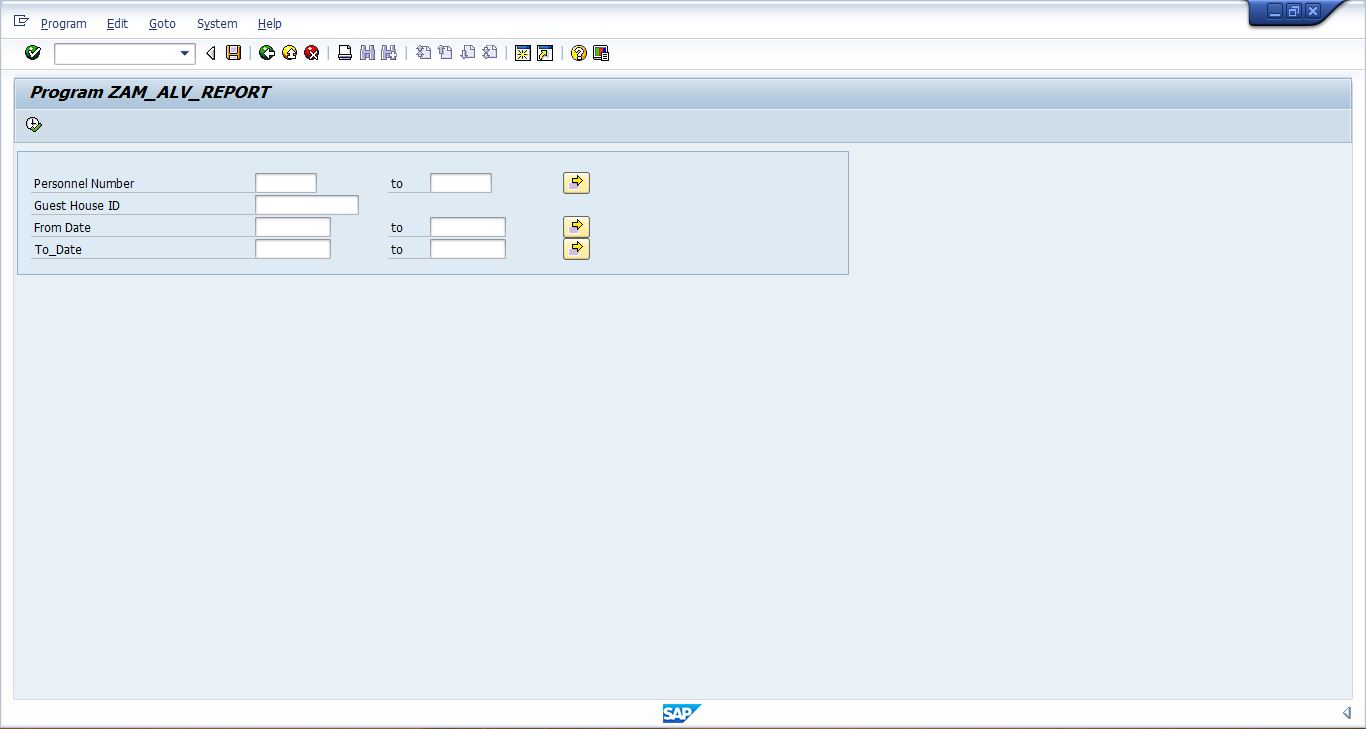
For the personnel number field, the user is provided with a search help.

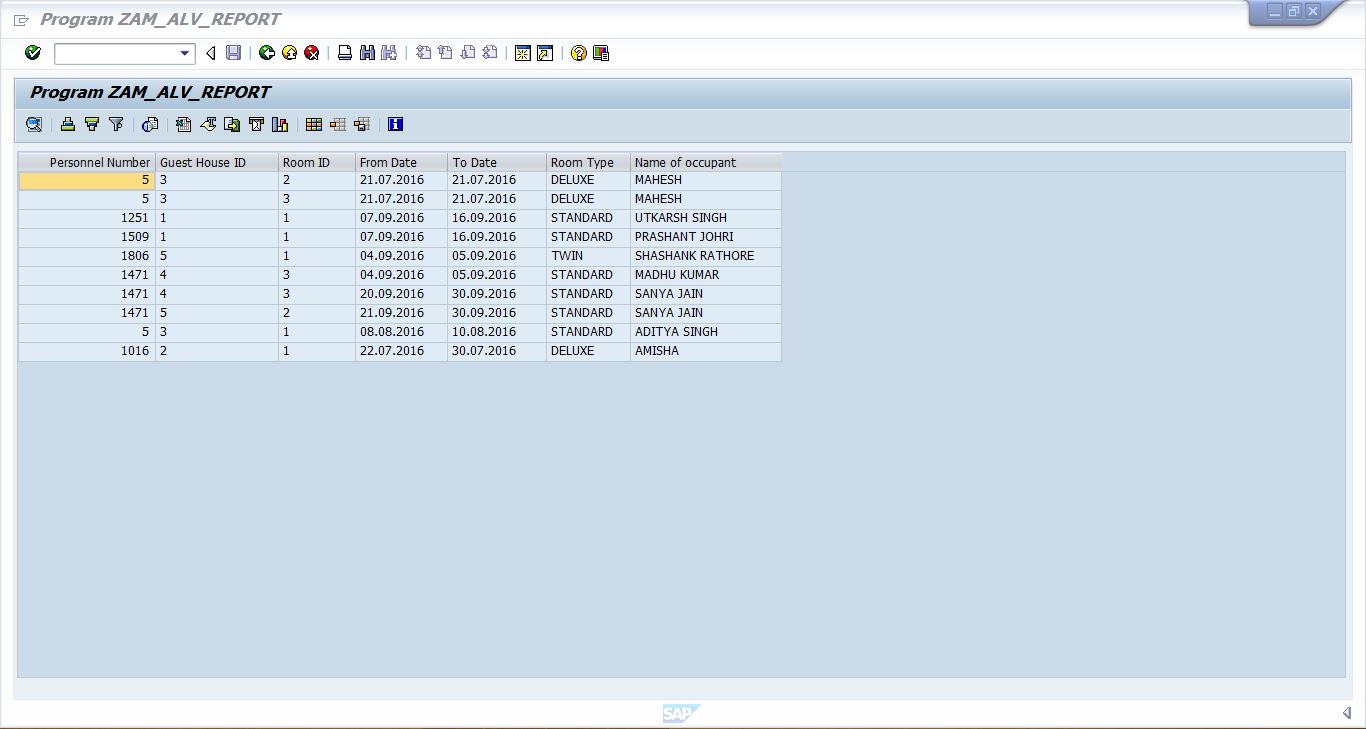
If the entered personnel number is invalid, the user is given an error message.

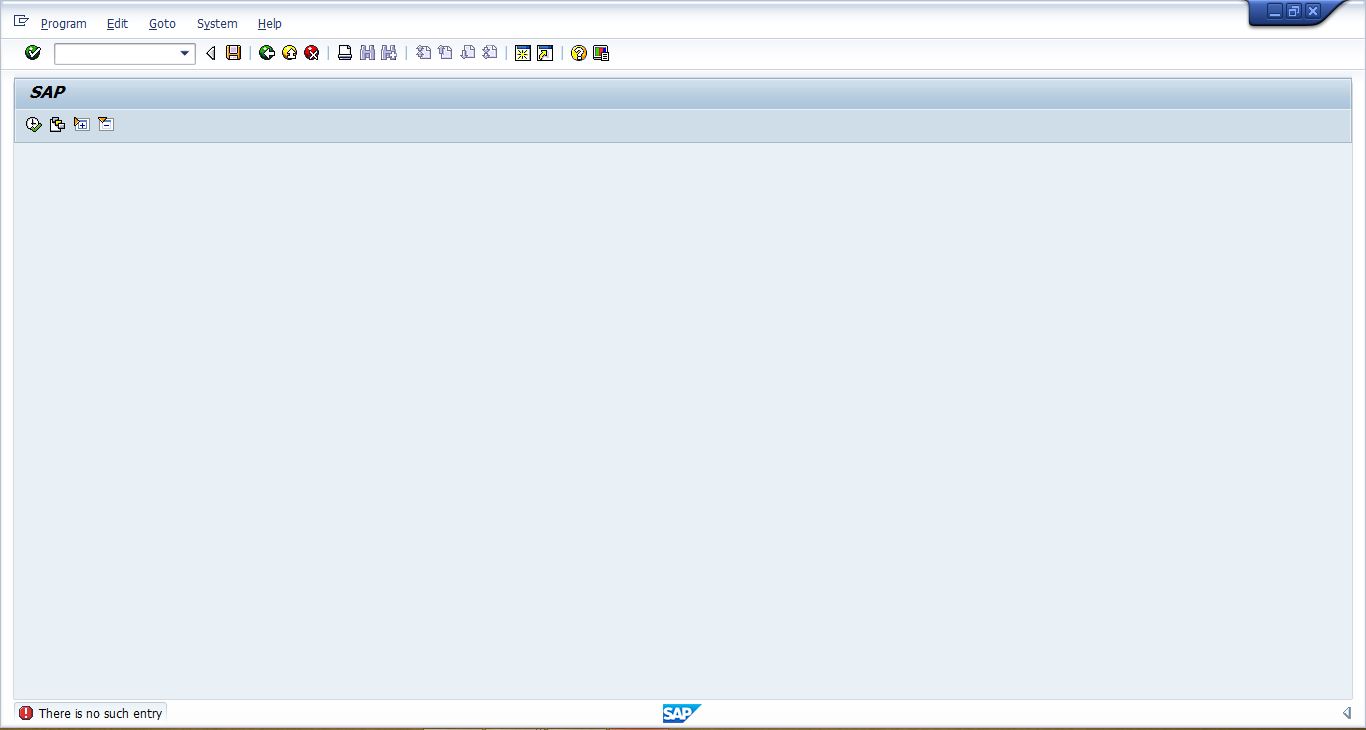
If the personnel number is valid and the room has been booked, the user is provided with a confirmation message.

**Generate report:**

If the user chooses to generate a report, the report ZAM\_ALV\_REPORT is called.



Upon execution of the report, the user is provided with an ALV containing the data corresponding to the search criteria.

If there is no record corresponding to the search criteria, the user is given an error message.

**VALIDATION CHECKS**

A common problem with computer system is that it is very easy to put incorrect data into them. So the input data is validated to minimize errors and data entry. For certain data specific code has been given and validations are done which enable the user to enter the required data and correct them if they have entered wrong codes, e.g. you could mistype a link name or URL in a database resulting in reports being occurred in the wrong link name. If you put incorrect data into the computer system, then you will get incorrect results out of it. Processing incorrect inputs will produce incorrect outputs. This leads to the acronym: GIGO (GARBAGE IN GARBAGE OUT). Sometimes incorrect data can actually cause a computer system to stop work temporarily. This is a particular problem in batch processing system when data may be processed overnights. If incorrect data stops a batch processing systems from working, then a whole night processing time may be lost. People who develop computer systems go to a lot of trouble to make it difficult for incorrect data to be entered. The two main techniques used for this purpose are:

> **VERIFICATION:**

The data entered by the user is verified before further execution of on the entered data.

To ensure that inconsistent and invalid data do not make it into the database and the programs, search helps, drop downs and required constraints were applied.

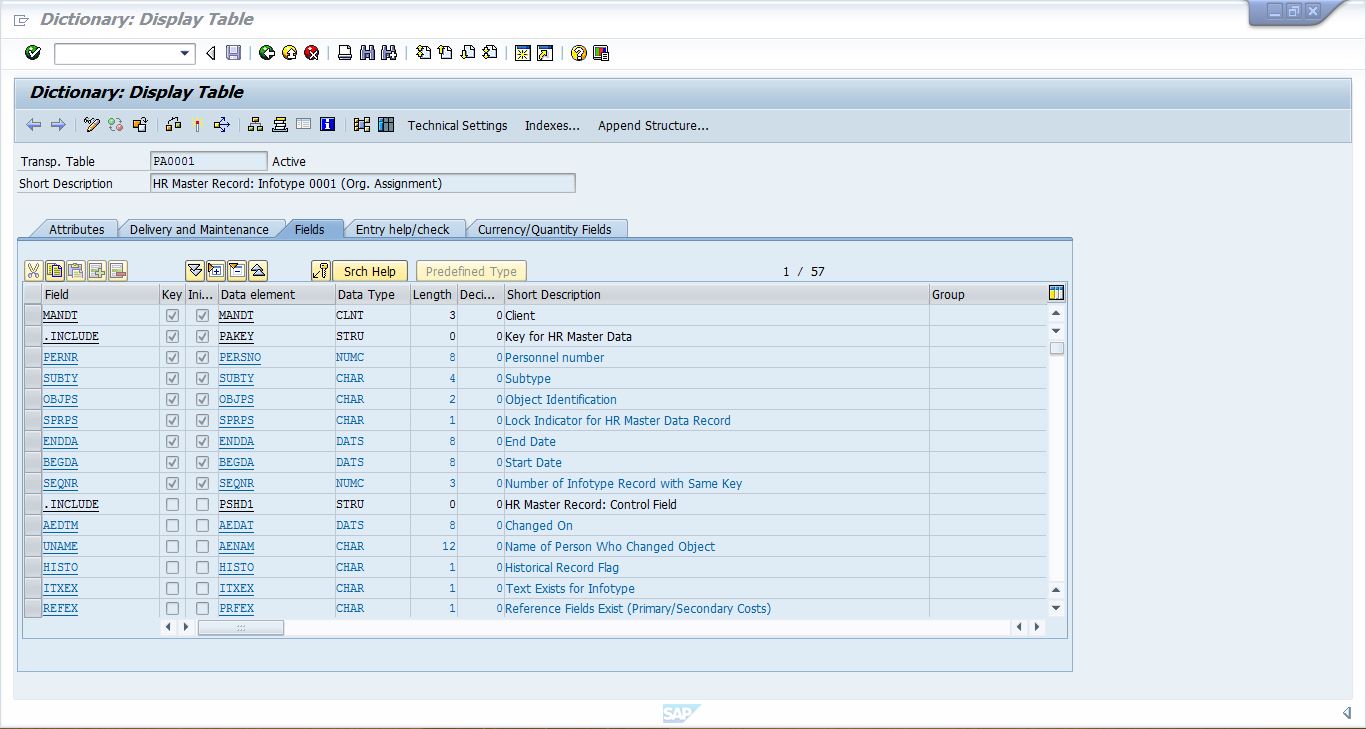
> **VALIDATION**

Before the data was used in any form, it was appropriately validated against existing database and in case of inconsistencies, appropriate error messages were returned.

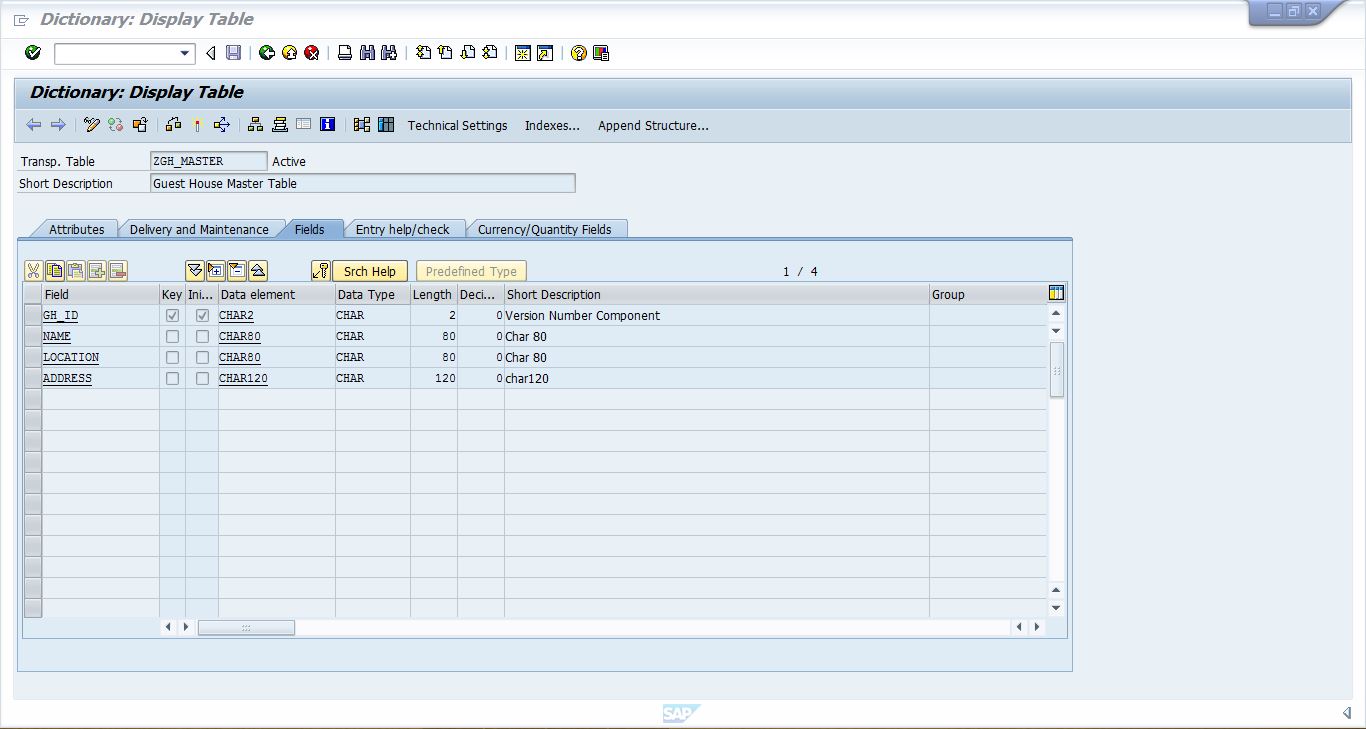
**TABLE OVERVIEW**

**TABLE: PA0001**

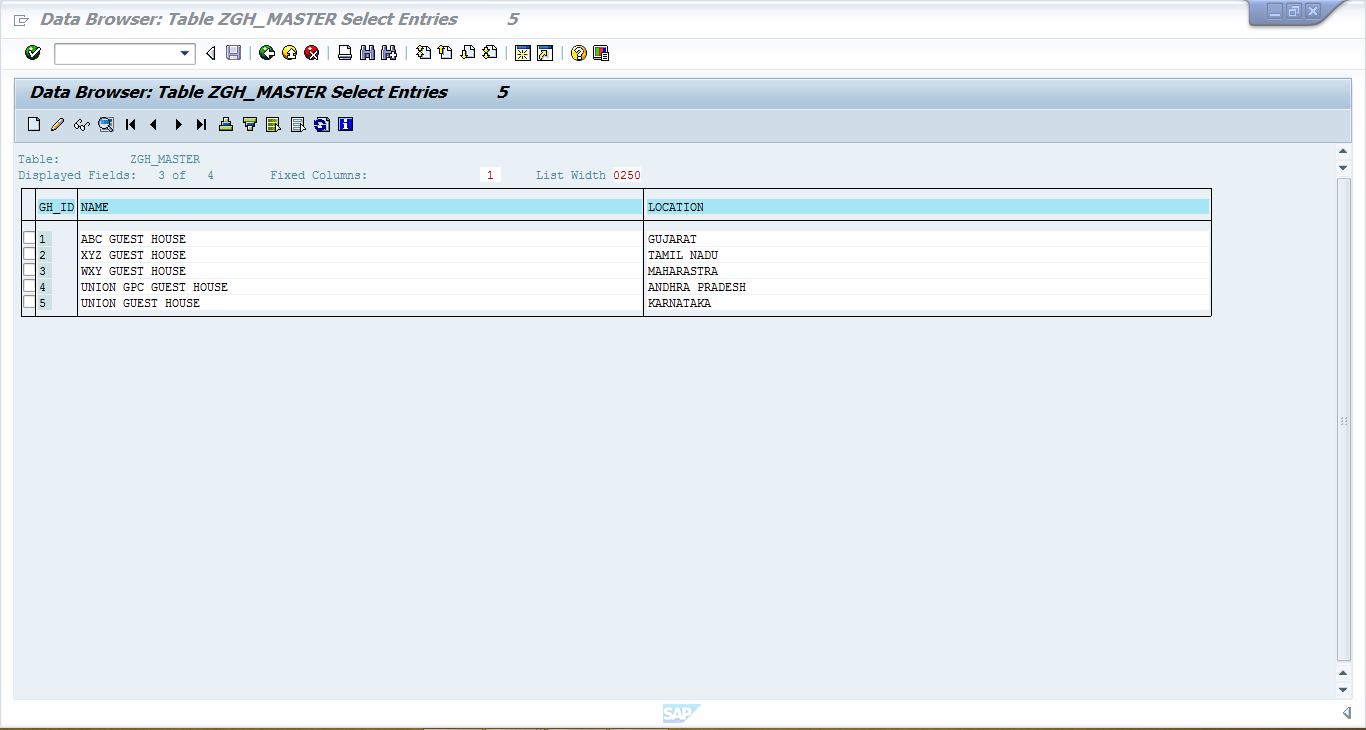
This pre-existing table was used to verify employee details.



**TABLE: zgh\_master**

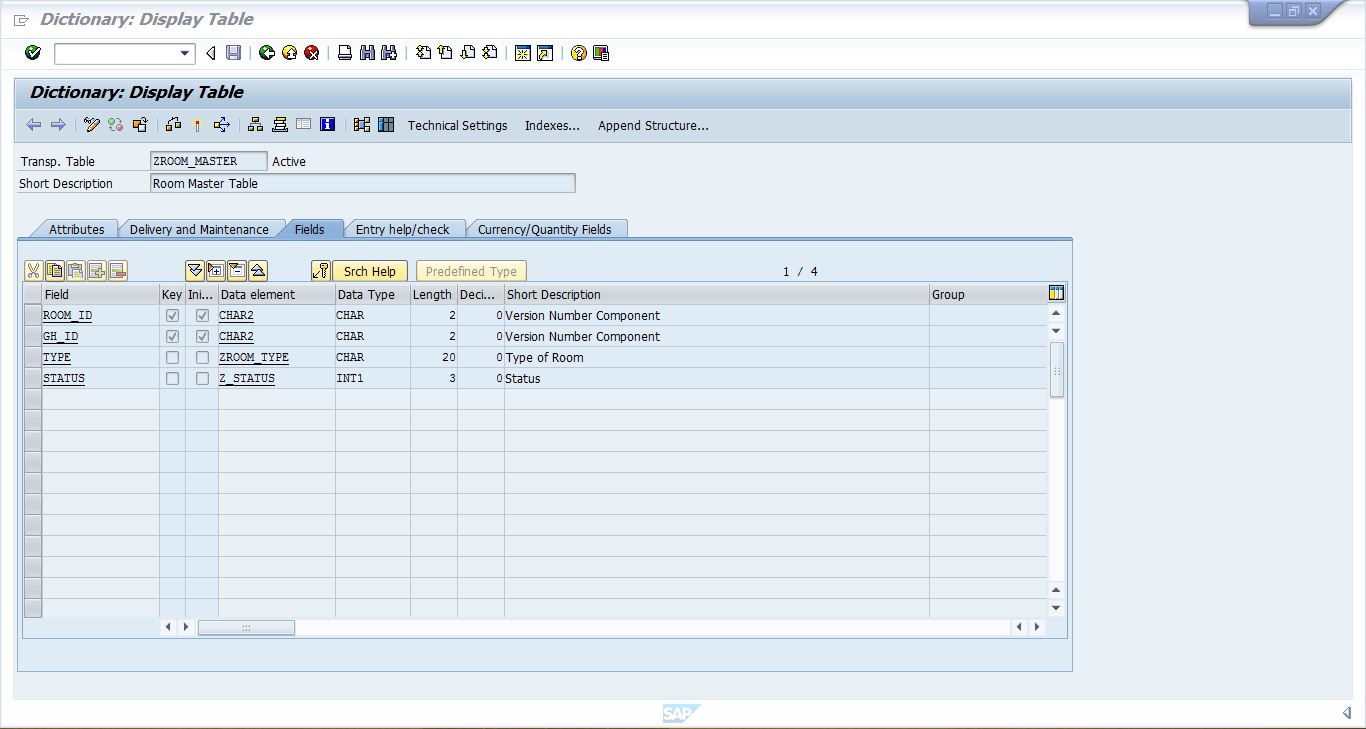
It is a transparent table containing the details of all the guesthouses.

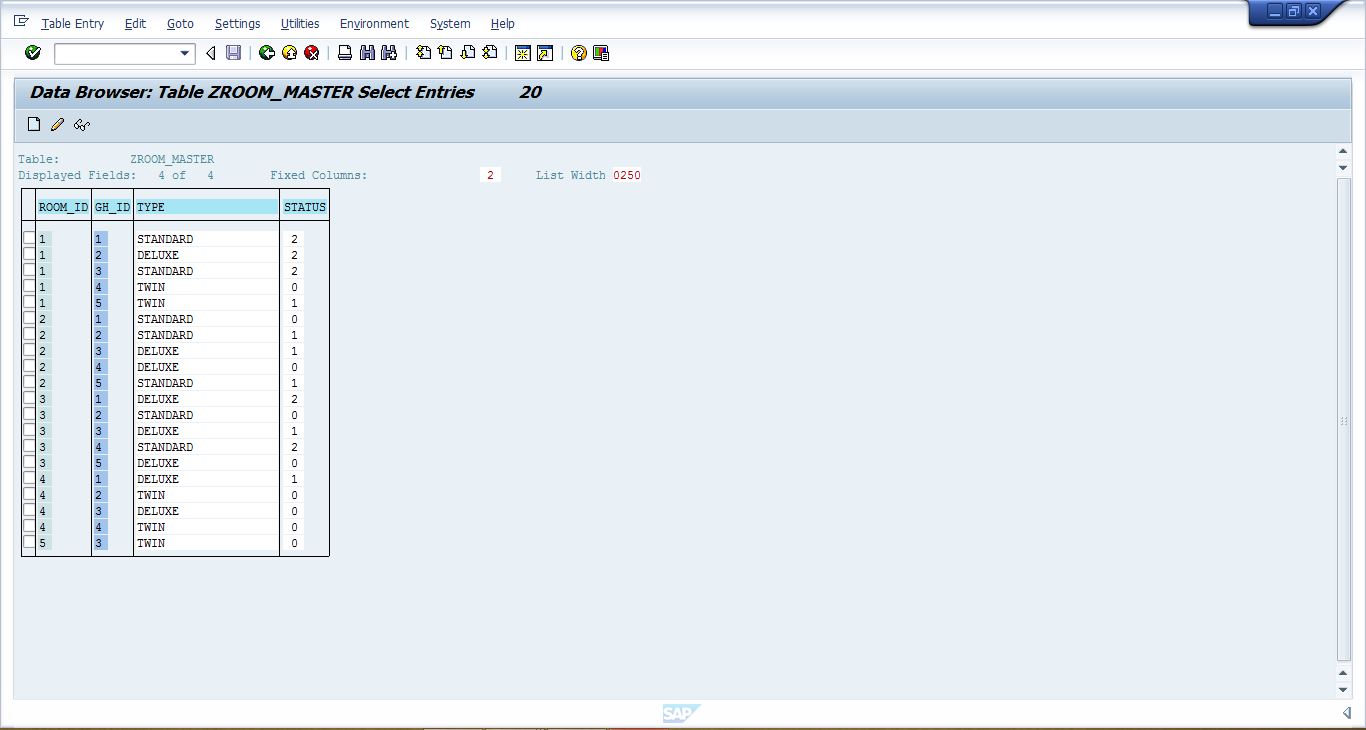
**Contents:**

****

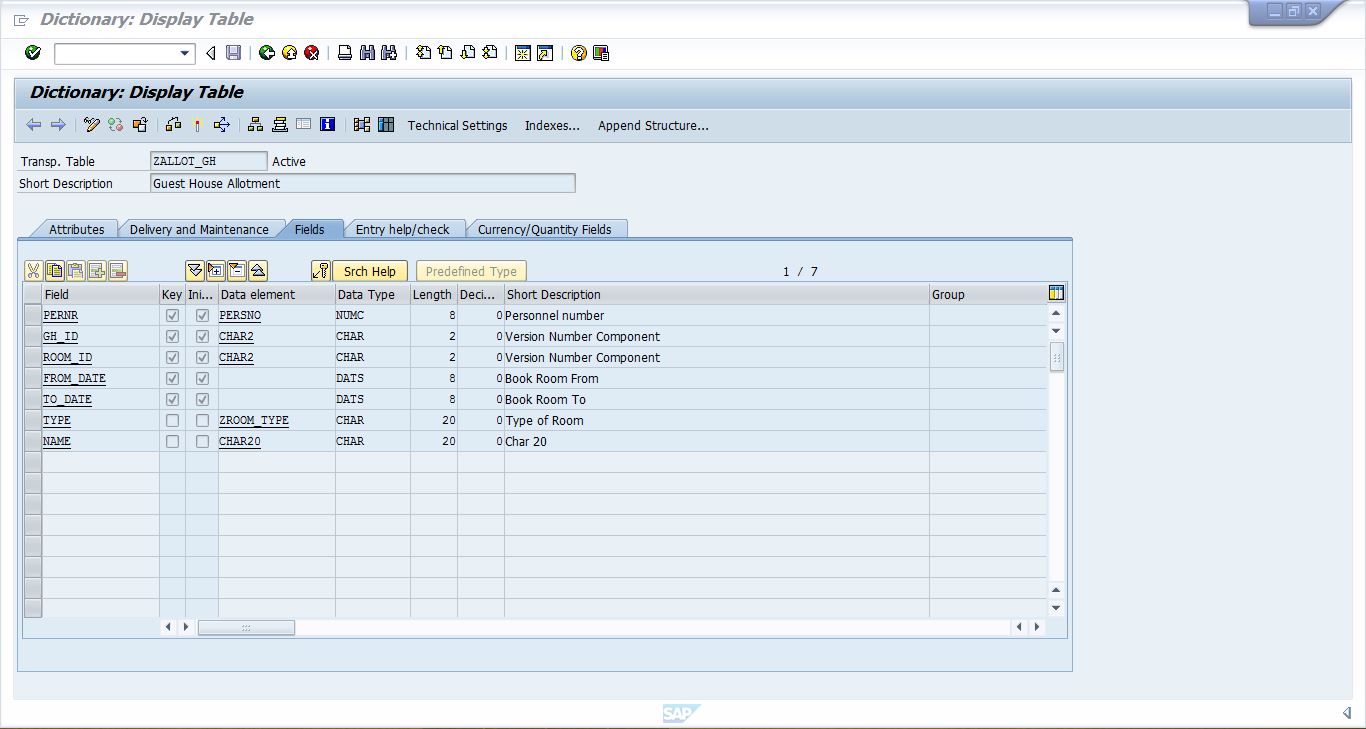
**TABLE: zroom\_master**

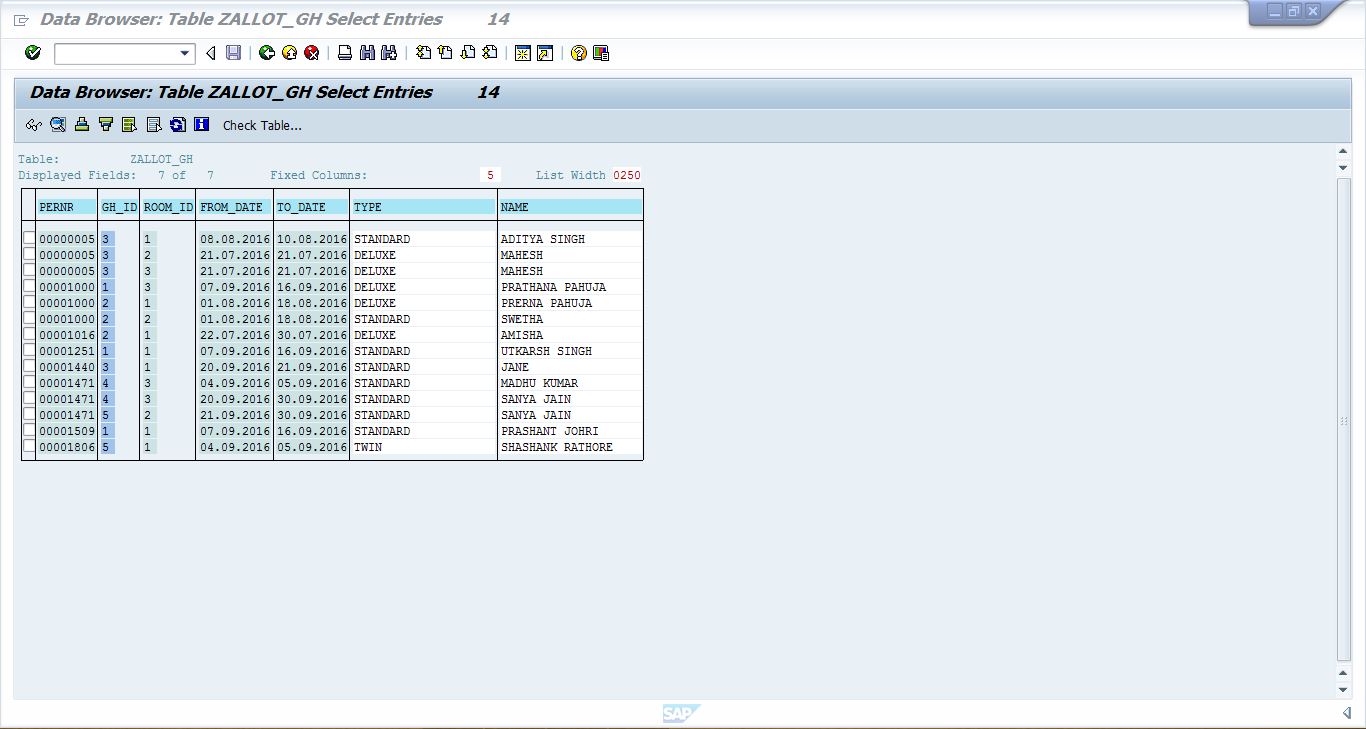
It is a transparent table and contains details of all the rooms in the guesthouses.



**Contents:**

**TABLE: zallot\_gh**

Initially empty, the table contains list of all booking made for the guesthouses.

**Contents:**

**PROGRAM CODE:**

**Screen 9000:**

PROCESS BEFORE OUTPUT.  
 MODULE STATUS\_9000.  
  
PROCESS AFTER INPUT.  
 MODULE USER\_COMMAND\_9000.

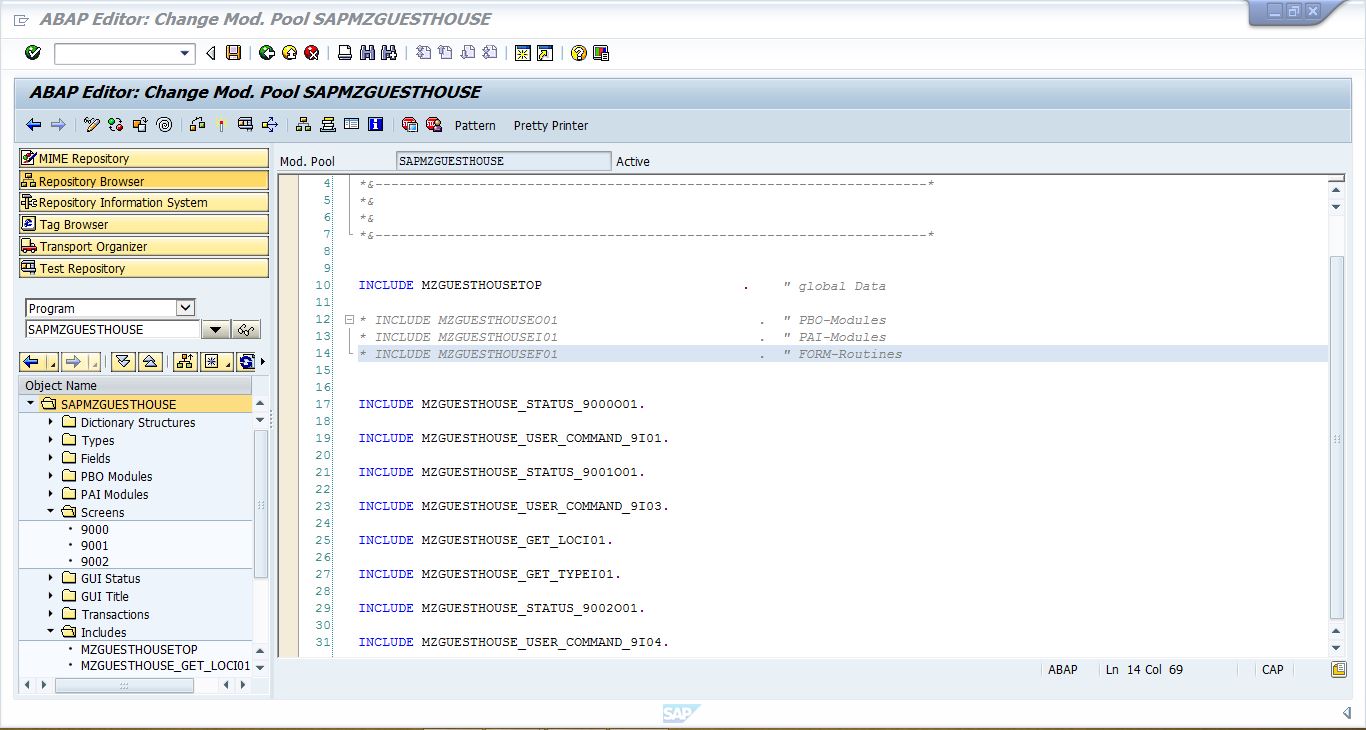
**Screen 0002:**

process before output.  
  module status\_9001.  
  
process after input.  
  module user\_command\_9001.  
  
process on value-request.  
  field rlocation module get\_loc.  
  field Rtype module get\_type.

**Screen 0003:**

PROCESS BEFORE OUTPUT.  
 MODULE STATUS\_9002.  
  
PROCESS AFTER INPUT.  
 MODULE USER\_COMMAND\_9002.

**The includes in the ABAP program are:**

****

**MZGUESTHOUSETOP:**

*\*&---------------------------------------------------------------------\**  
*\*& Include MZGUESTHOUSETOP                                   Module Pool      SAPMZGUESTHOUSE*  
*\*&*  
*\*&---------------------------------------------------------------------\**  
  
program  sapmzguesthouse.  
  
tables: zgh\_master, zroom\_master, zallot\_gh.  
  
types: begin of type\_loc,  
  location type zgh\_master-location,  
  end of type\_loc.  
  
data itab\_loc type standard table of type\_loc with header line.  
  
  
data: pno like pa0001-pernr,  
      txtpernr like zallot\_gh-pernr,  
      txtname like zallot\_gh-name.  
  
  
data : gid like zgh\_master-gh\_id,  
      rid like zroom\_master-room\_id value 0,  
      rlocation like zgh\_master-location,  
      rtype like zroom\_master-type.  
  
  
data : g\_t\_id type vrm\_id,  
       g\_t\_list type vrm\_values,  
       g\_t\_value like line of g\_t\_list.  
  
data wa\_loc like line of itab\_loc.  
  
types: begin of type\_allot,  
  room\_id type zroom\_master-room\_id,  
  end of type\_allot.  
  
  
data itab\_allot type standard table of type\_allot.  
data wa\_allot like line of itab\_allot.  
  
  
types: begin of type\_type,  
  type type zroom\_master-type,  
  end of type\_type.  
  
  
data itab\_type type standard table of type\_type with header line.  
  
*\*types: begin of type\_count,*  
*\*  gh\_id type zallot\_gh-gh\_id,*  
*\*  room\_id type zallot\_gh-room\_id,*  
*\*  count\_of type i,*  
*\*  end of type\_count.*  
  
*\*data itab\_count type standard table of type\_count.*  
*\*data wa\_status like line of itab\_count.*  
  
data : g\_t\_id2 type vrm\_id,  
       g\_t\_list2 type vrm\_values,  
       g\_t\_value2 like line of g\_t\_list2.  
  
data wa\_type like line of itab\_type.  
  
data line type zallot\_gh.  
  
data : ok\_code9000 like sy-ucomm,  
       ok\_code9001 like sy-ucomm,  
       ok\_code9002 like sy-ucomm.  
  
data : from\_dat type dats,  
       to\_dat type dats,  
       count\_of type i,  
       rcnt type i.

**PBO modules:**

*\*----------------------------------------------------------------------\**  
*\*\*\*INCLUDE MZGUESTHOUSE\_STATUS\_9000O01 .*  
*\*----------------------------------------------------------------------\**  
*\*&---------------------------------------------------------------------\**  
*\*&      Module  STATUS\_9000  OUTPUT*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
module STATUS\_9000 output.  
  SET PF-STATUS '9000'.  
  SET TITLEBAR '9000'.  
  
endmodule.                 *" STATUS\_9000  OUTPUT*

*\*&---------------------------------------------------------------------\**  
*\*&      Module  STATUS\_9001  OUTPUT*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
module STATUS\_9001 output.  
  SET PF-STATUS '9001'.  
  SET TITLEBAR '9001'.  
  
endmodule.                 *" STATUS\_9001  OUTPUT*  
*\*&---------------------------------------------------------------------\**  
*\*&      Module  STATUS\_9002  OUTPUT*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
module STATUS\_9002 output.  
  SET PF-STATUS '9002'.  
  SET TITLEBAR '9002'.  
  
endmodule.                 *" STATUS\_9002  OUTPUT*

**PAI modules:**

*\*----------------------------------------------------------------------\**  
*\*\*\*INCLUDE MZGUESTHOUSE\_GET\_LOCI01 .*  
*\*----------------------------------------------------------------------\**  
*\*&---------------------------------------------------------------------\**  
*\*&      Module  GET\_LOC  INPUT*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
module get\_loc input.  
  
  refresh : g\_t\_list.  
  clear : g\_t\_list.  
  
  refresh : itab\_loc.  
  
  
  select location from zgh\_master into corresponding fields of table itab\_loc.  
  
  loop at itab\_loc into wa\_loc.  
    clear g\_t\_value.  
    g\_t\_value-key  = wa\_loc-location.  
    g\_t\_value-text = wa\_loc-location.  
    append g\_t\_value to g\_t\_list.  
  endloop.  
  
  g\_t\_id = 'RLOCATION'.  
  
  call function 'VRM\_SET\_VALUES'  
    exporting  
      id     = g\_t\_id  
      values = g\_t\_list.  
  
endmodule.                 *" GET\_LOC  INPUT*

*\*----------------------------------------------------------------------\**  
*\*\*\*INCLUDE MZGUESTHOUSE\_GET\_TYPEI01 .*  
*\*----------------------------------------------------------------------\**  
*\*&---------------------------------------------------------------------\**  
*\*&      Module  GET\_TYPE  INPUT*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
module get\_type input.  
  
  refresh : g\_t\_list2.  
  clear : g\_t\_list2.  
  
  refresh : itab\_type.  
  
  
  select type from zroom\_master into corresponding fields of table itab\_type.  
  
  SORT itab\_type.  
  DELETE ADJACENT DUPLICATES FROM itab\_type COMPARING type.  
  
  loop at itab\_type into wa\_type.  
    clear g\_t\_value2.  
    g\_t\_value2-key  = wa\_type-type.  
    g\_t\_value2-text = wa\_type-type.  
    append g\_t\_value2 to g\_t\_list2.  
  endloop.  
  
  g\_t\_id2 = 'RTYPE'.  
  
  call function 'VRM\_SET\_VALUES'  
    exporting  
      id     = g\_t\_id2  
      values = g\_t\_list2.  
  
endmodule.                 *" GET\_TYPE  INPUT*  
*\*----------------------------------------------------------------------\**  
*\*\*\*INCLUDE MZGUESTHOUSE\_USER\_COMMAND\_9I01 .*  
*\*----------------------------------------------------------------------\**  
*\*&---------------------------------------------------------------------\**  
*\*&      Module  USER\_COMMAND\_9000  INPUT*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
module USER\_COMMAND\_9000 input.  
  
  CASE OK\_CODE9000.  
  
    WHEN 'EXIT'.  
      LEAVE PROGRAM.  
    WHEN 'ZBOOK'.  
      CALL SCREEN '9001'.  
    WHEN 'ZRT'.  
      SUBMIT ZAM\_ALV\_REPORT VIA selection-screen.  
    *"WHEN 'ZCHECK'.*  
     *" CALL SCREEN '9003'.*  
  
  ENDCASE.  
  
endmodule.                 *" USER\_COMMAND\_9000  INPUT*  
*\*----------------------------------------------------------------------\**  
*\*\*\*INCLUDE MZGUESTHOUSE\_USER\_COMMAND\_9I03 .*  
*\*----------------------------------------------------------------------\**  
*\*&---------------------------------------------------------------------\**  
*\*&      Module  USER\_COMMAND\_9001  INPUT*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
module user\_command\_9001 input.  
  
  case ok\_code9001.  
    when 'EXIT'.  
      leave program.  
    when 'BACK'.  
      call screen '9000'.  
    when 'ZCA'.  
      select single gh\_id from zgh\_master into gid where location eq rlocation.  
      select single room\_id from zroom\_master into rid where gh\_id eq gid and type eq rtype and status = 0.  
      if sy-subrc = 0.  
        call screen '9002'.  
      endif.  
  
      *"if rid eq 0.*  
        select room\_id from zroom\_master into corresponding fields of table itab\_allot where gh\_id eq gid and type eq rtype.  
        loop at itab\_allot into wa\_allot.  
          select count(\*) from zallot\_gh into count\_of where ( from\_date gt to\_dat or to\_date lt from\_dat ) and gh\_id = gid and room\_id = wa\_allot-room\_id.  
          select status from zroom\_master into rcnt where gh\_id = gid and room\_id = wa\_allot-room\_id.  
          endselect.  
          if count\_of = rcnt.  
            rid = wa\_allot-room\_id.  
            call screen '9002'.  
          endif.  
        endloop.  
  
        message i000(zhr) with 'Requested room unavailable. Try changing the dates'.  
  
      *"endif.*  
  
*\*        select gh\_id room\_id count(\*) as count\_of into itab\_count from itab\_allot group by room\_id.*  
*\*        DESCRIBE TABLE itab\_allot LINES count\_of.*  
  
  
*\*         loop at itab\_count into wa\_status.*  
*\*            select room\_id from zroom\_master into rid where room\_id = wa\_status-room\_id and gh\_id = wa\_status-gh\_id and count\_of = wa\_status-status.*  
*\*            endloop.*  
  
    endcase.  
  
    endmodule.                 *" USER\_COMMAND\_9001  INPUT*  
*\*----------------------------------------------------------------------\**  
*\*\*\*INCLUDE MZGUESTHOUSE\_USER\_COMMAND\_9I04 .*  
*\*----------------------------------------------------------------------\**  
*\*&---------------------------------------------------------------------\**  
*\*&      Module  USER\_COMMAND\_9002  INPUT*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
module user\_command\_9002 input.  
  
  case ok\_code9002.  
  
    when 'EXIT'.  
      leave program.  
    when 'BACK'.  
      call screen '9001'.  
    when 'ZRCON'.  
      select pernr from pa0001 into pno where pernr = txtpernr.  
      endselect.  
      if pno eq txtpernr.  
        line-name = txtname.  
        line-pernr = txtpernr.  
        line-gh\_id = gid.  
        line-room\_id = rid.  
        line-type = rtype.  
        line-from\_date = from\_dat.  
        line-to\_date = to\_dat.  
        insert into zallot\_gh values line.  
        if sy-subrc = 0.  
          update zroom\_master set status = status + 1 where gh\_id = gid and room\_id = rid.  
          message i000(zhr) with 'Record Updated Successfully'.  
        endif.  
      endif.  
  
      if pno ne txtpernr.  
        message e000(zhr) with 'Enter a valid Personnel No.'.  
      endif.  
  
    endcase.  
  
  endmodule.                 *" USER\_COMMAND\_9002  INPUT*

**ZAM\_ALV\_REPORT:**

**Include:**

*\*&---------------------------------------------------------------------\**  
*\*& Report  ZAM\_ALV\_REPORT*  
*\*&*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&*  
*\*&---------------------------------------------------------------------\**  
  
INCLUDE ZAM\_ALV\_REPORT\_TOP                      .    *" global Data*  
  
*\* INCLUDE ZAM\_ALV\_REPORT\_O01                      .  " PBO-Modules*  
*\* INCLUDE ZAM\_ALV\_REPORT\_I01                      .  " PAI-Modules*  
 INCLUDE ZAM\_ALV\_REPORT\_F01                      .  *" FORM-Routines*  
  
 INITIALIZATION.  
  
 At selection-screen.  
  
start-of-selection.  
   perform f\_get\_data.  
  
end-of-selection.  
  if not it\_allot is initial.  
    perform f\_fieldcat.  
  
    perform f\_display.  
  endif.  
  
  endmodule.                 *" USER\_COMMAND\_9002  INPUT*

**ZAM\_ALV\_REPORT\_TOP:**

*\*&---------------------------------------------------------------------\**  
*\*& Include ZAM\_ALV\_REPORT\_TOP                                Report ZAM\_ALV\_REPORT*  
*\*&*  
*\*&---------------------------------------------------------------------\**  
  
REPORT   ZAM\_ALV\_REPORT.  
  
type-pools : slis.  
  
data : gs\_allot type zallot\_gh.  
data : count\_of type i.  
  
selection-screen begin of block b1 with frame.  
  
  select-options : s\_pernr for gs\_allot-pernr.  
  parameters : s\_gid type i.  
  select-options : s\_fdate for gs\_allot-from\_date.  
  select-options : s\_tdate for gs\_allot-to\_date.  
  
selection-screen end of block b1.  
  
types : begin of ty\_allot,  
        pernr type persno,  
        gh\_id type char2,  
        room\_id type char2,  
        from\_date type dats,  
        to\_date type dats,  
        type type zroom\_type,  
        name type char20,  
        end of ty\_allot.  
  
data : it\_allot type table of ty\_allot,  
       wa\_allot type          ty\_allot,  
  
       it\_fcat type slis\_t\_fieldcat\_alv,  
       wa\_fcat type slis\_fieldcat\_alv.

**ZAM\_ALV\_REPORT\_F01:**

*\*&---------------------------------------------------------------------\**  
*\*&  Include           ZAM\_ALV\_REPORT\_F01*  
*\*&---------------------------------------------------------------------\**  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_GET\_DATA*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
*\*  -->  p1        text*  
*\*  <--  p2        text*  
*\*----------------------------------------------------------------------\**  
form f\_get\_data .  
  refresh : it\_allot.  
  clear : wa\_allot.  
  
  if not s\_pernr is initial. *"pernr initialised*  
    if not s\_gid is initial. *"gh\_id and pernr initialised*  
      if not s\_fdate is initial. *"fdate initialised*  
        if not s\_tdate is initial. *"tdate initialised*  
          select pernr  
            gh\_id  
            room\_id  
            from\_date  
            to\_date  
            type  
            name  
          from zallot\_gh  
          into table it\_allot  
          where pernr in s\_pernr and gh\_id = s\_gid and from\_date in s\_fdate and to\_date in s\_tdate.  
            if sy-subrc <> 0.  
              message e000(zhr) with 'There is no such entry'.  
            endif.  
  
        else.                 *"all but tdate*  
          select pernr  
            gh\_id  
            room\_id  
            from\_date  
            to\_date  
            type  
            name  
          from zallot\_gh  
          into table it\_allot  
          where pernr in s\_pernr and gh\_id = s\_gid and from\_date in s\_fdate.  
            if sy-subrc <> 0.  
              message e000(zhr) with 'There is no such entry'.  
            endif.  
  
        endif.  
  
      else.                          *"fdate not initialised, gh\_id and pernr init*  
        if not s\_tdate is initial.  
        select pernr  
          gh\_id  
          room\_id  
          from\_date  
          to\_date  
          type  
          name  
        from zallot\_gh  
        into table it\_allot  
        where pernr in s\_pernr and gh\_id = s\_gid and to\_date in s\_tdate.  
          if sy-subrc <> 0.  
      message e000(zhr) with 'There is no such entry'.  
    endif.  
  
        else.             *"fdate and tdate not init*  
          select pernr  
          gh\_id  
          room\_id  
          from\_date  
          to\_date  
          type  
          name  
        from zallot\_gh  
        into table it\_allot  
        where pernr in s\_pernr and gh\_id = s\_gid.  
            if sy-subrc <> 0.  
      message e000(zhr) with 'There is no such entry'.  
    endif.  
  
        endif.  
      endif.  
  
*"---------------------------------next set-------------------------------------------------------*  
  
    else.                    *"gh\_id not initialised*  
      if not s\_fdate is initial.  
        if not s\_tdate is initial. *"all but gh\_id*  
           select pernr  
                 gh\_id  
                 room\_id  
                 from\_date  
                 to\_date  
                 type  
                 name  
                 from zallot\_gh  
                 into table it\_allot  
                 where pernr in s\_pernr and from\_date in s\_fdate and to\_date in s\_tdate.  
               if sy-subrc <> 0.  
                     message e000(zhr) with 'There is no such entry'.  
               endif.  
  
         else.                 *"all but gh\_id and tdate*  
          select pernr  
             gh\_id  
             room\_id  
             from\_date  
             to\_date  
             type  
             name  
            from zallot\_gh  
            into table it\_allot  
            where pernr in s\_pernr and from\_date in s\_fdate.  
            if sy-subrc <> 0.  
               message e000(zhr) with 'There is no such entry'.  
            endif.  
         endif.  
  
      else. *"fdate not initialised*  
         if not s\_tdate is initial. *"fdate and gh\_id NI*  
           select pernr  
                 gh\_id  
                 room\_id  
                 from\_date  
                 to\_date  
                 type  
                 name  
                from zallot\_gh  
               into table it\_allot  
               where pernr in s\_pernr and to\_date in s\_tdate.  
               if sy-subrc <> 0.  
                     message e000(zhr) with 'There is no such entry'.  
               endif.  
  
         else.                 *"only pernr initialised.*  
          select pernr  
             gh\_id  
             room\_id  
             from\_date  
             to\_date  
             type  
             name  
            from zallot\_gh  
            into table it\_allot  
            where pernr in s\_pernr.  
            if sy-subrc <> 0.  
               message e000(zhr) with 'There is no such entry'.  
            endif.  
        endif.  
  
       endif.  
  
    endif.  
  
  else.  
    if not s\_gid is initial.    *"gh\_id initialised but pernr not initialised.*  
    if not s\_fdate is initial. *"fdate initialised*  
        if not s\_tdate is initial. *"tdate initialised*  
          select pernr  
            gh\_id  
            room\_id  
            from\_date  
            to\_date  
            type  
            name  
          from zallot\_gh  
          into table it\_allot  
          where gh\_id = s\_gid and from\_date in s\_fdate and to\_date in s\_tdate.  
            if sy-subrc <> 0.  
              message e000(zhr) with 'There is no such entry'.  
            endif.  
  
        else.                 *"all but tdate and pernr*  
          select pernr  
            gh\_id  
            room\_id  
            from\_date  
            to\_date  
            type  
            name  
          from zallot\_gh  
          into table it\_allot  
          where gh\_id = s\_gid and from\_date in s\_fdate.  
            if sy-subrc <> 0.  
              message e000(zhr) with 'There is no such entry'.  
              endif.  
  
        endif.  
  
      else.                          *"fdate not initialised, gh\_id init*  
        if not s\_tdate is initial.  
        select pernr  
          gh\_id  
          room\_id  
          from\_date  
          to\_date  
          type  
          name  
        from zallot\_gh  
        into table it\_allot  
        where gh\_id = s\_gid and to\_date in s\_tdate.  
          if sy-subrc <> 0.  
            message e000(zhr) with 'There is no such entry'.  
          endif.  
  
        else.             *"fdate and tdate not init*  
          select pernr  
          gh\_id  
          room\_id  
          from\_date  
          to\_date  
          type  
          name  
        from zallot\_gh  
        into table it\_allot  
        where gh\_id = s\_gid.  
            if sy-subrc <> 0.  
                message e000(zhr) with 'There is no such entry'.  
            endif.  
  
        endif.  
        endif.  
  
  
  else.                    *"gh\_id not initialised*  
      if not s\_fdate is initial.  
        if not s\_tdate is initial. *"all but gh\_id*  
           select pernr  
                 gh\_id  
                 room\_id  
                 from\_date  
                 to\_date  
                 type  
                 name  
                from zallot\_gh  
               into table it\_allot  
               where from\_date in s\_fdate and to\_date in s\_tdate.  
               if sy-subrc <> 0.  
                     message e000(zhr) with 'There is no such entry'.  
               endif.  
  
          else.                 *"only fdate*  
          select pernr  
             gh\_id  
             room\_id  
             from\_date  
             to\_date  
             type  
             name  
            from zallot\_gh  
            into table it\_allot  
            where from\_date in s\_fdate.  
            if sy-subrc <> 0.  
               message e000(zhr) with 'There is no such entry'.  
            endif.  
         endif.  
  
         else. *"fdate not initialised*  
           if not s\_tdate is initial. *"only tdate*  
           select pernr  
                 gh\_id  
                 room\_id  
                 from\_date  
                 to\_date  
                 type  
                 name  
                from zallot\_gh  
               into table it\_allot  
               where to\_date in s\_tdate.  
               if sy-subrc <> 0.  
                     message e000(zhr) with 'There is no such entry'.  
               endif.  
          else.  
            select pernr  
              gh\_id  
              room\_id  
              from\_date  
              to\_date  
              type  
              name  
              from zallot\_gh  
              into table it\_allot  
              up to 10 rows.  
           endif.  
         endif.  
          endif.  
         endif.  
  
endform.                    *" F\_GET\_DATA*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_FIELDCAT*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
*\*  -->  p1        text*  
*\*  <--  p2        text*  
*\*----------------------------------------------------------------------\**  
form f\_fieldcat .  
  refresh : it\_fcat.  
  clear : wa\_fcat.  
  
  wa\_fcat-row\_pos = '1'.  
  wa\_fcat-col\_pos = '1'.  
  wa\_fcat-fieldname = 'PERNR'.  
  wa\_fcat-tabname = 'it\_allot'.  
  wa\_fcat-seltext\_m = text-001.  
  append wa\_fcat to it\_fcat.  
  clear: wa\_fcat.  
  
  wa\_fcat-row\_pos = '1'.  
  wa\_fcat-col\_pos = '2'.  
  wa\_fcat-fieldname = 'GH\_ID'.  
  wa\_fcat-tabname = 'it\_allot'.  
  wa\_fcat-seltext\_m = text-002.  
  append wa\_fcat to it\_fcat.  
  clear: wa\_fcat.  
  
  wa\_fcat-row\_pos = '1'.  
  wa\_fcat-col\_pos = '3'.  
  wa\_fcat-fieldname = 'ROOM\_ID'.  
  wa\_fcat-tabname = 'it\_allot'.  
  wa\_fcat-seltext\_m = text-003.  
  append wa\_fcat to it\_fcat.  
  clear: wa\_fcat.  
  
  wa\_fcat-row\_pos = '1'.  
  wa\_fcat-col\_pos = '4'.  
  wa\_fcat-fieldname = 'FROM\_DATE'.  
  wa\_fcat-tabname = 'it\_allot'.  
  wa\_fcat-seltext\_m = text-004.  
  append wa\_fcat to it\_fcat.  
  clear: wa\_fcat.  
  
  wa\_fcat-row\_pos = '1'.  
  wa\_fcat-col\_pos = '5'.  
  wa\_fcat-fieldname = 'TO\_DATE'.  
  wa\_fcat-tabname = 'it\_allot'.  
  wa\_fcat-seltext\_m = text-005.  
  append wa\_fcat to it\_fcat.  
  clear: wa\_fcat.  
  
  wa\_fcat-row\_pos = '1'.  
  wa\_fcat-col\_pos = '6'.  
  wa\_fcat-fieldname = 'TYPE'.  
  wa\_fcat-tabname = 'it\_allot'.  
  wa\_fcat-seltext\_m = text-006.  
  append wa\_fcat to it\_fcat.  
  clear: wa\_fcat.  
  
  wa\_fcat-row\_pos = '1'.  
  wa\_fcat-col\_pos = '7'.  
  wa\_fcat-fieldname = 'NAME'.  
  wa\_fcat-tabname = 'it\_allot'.  
  wa\_fcat-seltext\_m = text-007.  
  append wa\_fcat to it\_fcat.  
  clear: wa\_fcat.  
  
endform.                    *" F\_FIELDCAT*  
*\*&---------------------------------------------------------------------\**  
*\*&      Form  F\_DISPLAY*  
*\*&---------------------------------------------------------------------\**  
*\*       text*  
*\*----------------------------------------------------------------------\**  
*\*  -->  p1        text*  
*\*  <--  p2        text*  
*\*----------------------------------------------------------------------\**  
form f\_display .  
  
  
    call function 'REUSE\_ALV\_GRID\_DISPLAY'  
   exporting  
*\*                                           I\_INTERFACE\_CHECK                 = ' '*  
*\*                                           I\_BYPASSING\_BUFFER                = ' '*  
*\*                                           I\_BUFFER\_ACTIVE                   = ' '*  
     i\_callback\_program                = sy-repid  
*\*                                           I\_CALLBACK\_PF\_STATUS\_SET          = ' '*  
*\*                                           I\_CALLBACK\_USER\_COMMAND           = ' '*  
*\*                                           I\_CALLBACK\_TOP\_OF\_PAGE            = ' '*  
*\*                                           I\_CALLBACK\_HTML\_TOP\_OF\_PAGE       = ' '*  
*\*                                           I\_CALLBACK\_HTML\_END\_OF\_LIST       = ' '*  
*\*                                           I\_STRUCTURE\_NAME                  =*  
*\*                                           I\_BACKGROUND\_ID                   = ' '*  
*\*                                           I\_GRID\_TITLE                      =*  
*\*                                           I\_GRID\_SETTINGS                   =*  
*\*                                           IS\_LAYOUT                         =*  
     it\_fieldcat                       = it\_fcat  
*\*                                           IT\_EXCLUDING                      =*  
*\*                                           IT\_SPECIAL\_GROUPS                 =*  
*\*                                           IT\_SORT                           =*  
*\*                                           IT\_FILTER                         =*  
*\*                                           IS\_SEL\_HIDE                       =*  
*\*                                           I\_DEFAULT                         = 'X'*  
     i\_save                            = 'A'  
*\*                                           IS\_VARIANT                        =*  
*\*                                           IT\_EVENTS                         =*  
*\*                                           IT\_EVENT\_EXIT                     =*  
*\*                                           IS\_PRINT                          =*  
*\*                                           IS\_REPREP\_ID                      =*  
*\*                                           I\_SCREEN\_START\_COLUMN             = 0*  
*\*                                           I\_SCREEN\_START\_LINE               = 0*  
*\*                                           I\_SCREEN\_END\_COLUMN               = 0*  
*\*                                           I\_SCREEN\_END\_LINE                 = 0*  
*\*                                           I\_HTML\_HEIGHT\_TOP                 = 0*  
*\*                                           I\_HTML\_HEIGHT\_END                 = 0*  
*\*                                           IT\_ALV\_GRAPHICS                   =*  
*\*                                           IT\_HYPERLINK                      =*  
*\*                                           IT\_ADD\_FIELDCAT                   =*  
*\*                                           IT\_EXCEPT\_QINFO                   =*  
*\*                                           IR\_SALV\_FULLSCREEN\_ADAPTER        =*  
*\*                                         IMPORTING*  
*\*                                           E\_EXIT\_CAUSED\_BY\_CALLER           =*  
*\*                                           ES\_EXIT\_CAUSED\_BY\_USER            =*    tables  
      t\_outtab                          = it\_allot  
   exceptions  
     program\_error                     = 1  
     others                            = 2  
            .  
  if sy-subrc <> 0.  
*\* Implement suitable error handling here*  
  endif.  
  refresh : it\_allot.  
  
endform.                    *" F\_DISPLAY*

**FUTURE SCOPE:**

**This project can be further extended and improved by including other features like:**

* Following the hierarchy of the corporation, that is, multiple levels of administrators for various cores of the company.
* More items can be added to the database as per the requirements.
* More employees can be added to the database.
* Employees belonging to a specific core can be grouped together labelled as a separate core.

**REFERENCES:**

**SR NO. LINK**

1 www.saptechnical.com

2 www.abapmaster.com

3 www.sap-basis-abap.com

4 sap-abap-material.pdf

5 Sam’s Teach Yourself ABAP 4 in 21 days

6 Black-Book-for-ABAP

7 ABAP Documentation