

**HOTEL MANAGEMENT SOFTWARE**

**Documentation**

|  |  |
| --- | --- |
| Lớp: K6CDA01 - Nhóm:4 | |
| **Group Member** | * Nguyễn Mạnh Quang * Nguyễn Tiến Khánh * Dương Văn Hà * Trần Văn Tín |
| **Instructor** | Đinh Ngọc Lân |

**INDEX**

[1.Problem Definition 3](#_Toc478040283)

[1.1. Problem Abstraction 3](#_Toc478040284)

[1.2. The Current System 3](#_Toc478040286)

[1.3. The Proposed System 4](#_Toc478040287)

[1.4. Boundaries of the System 5](#_Toc478040288)

[1.5. Hardware and Software Requirements 5](#_Toc478040289)

[*1.5.1. Minimum Requirements* 5](#_Toc478040290)

[1.4.1 1.5.2. Recommended Requirements 6](#_Toc478040291)

[2. Customer Requirements Specification 6](#_Toc478040292)

[2.1. Users of the System 6](#_Toc478040293)

[2.2. System functions 6](#_Toc478040294)

[2.2.1. User 6](#_Toc478040295)

[2.2.3. Admin 9](#_Toc478040296)

[*3.System Designs* 13](#_Toc478040297)

[3.1 Architecture 13](#_Toc478040298)

[3.2. Data Flow Data (DFD) 14](#_Toc478040299)

[3.3 Flow Chart 22](#_Toc478040300)

[3.4. Entity Relationship Diagram 28](#_Toc478040301)

[3.5. Database Structure 29](#_Toc478040302)

[3.5.1. Database Diagram 30](#_Toc478040303)

[3.5.2 Table of database 31](#_Toc478040304)

[3.6 Screen Design 38](#_Toc478040305)

[*4.* *Checklists* 38](#_Toc478040306)

[4.1 Check List of Validation 38](#_Toc478040307)

[4.2 Submission Checklist 38](#_Toc478040308)

# Problem Definition

## 1.1. Problem Abstraction

## Along with the continuous development of computer technology and electronic networks, information technology has gained many great achievements, greatly contributed to the construction and social development of mankind. Today, Information Technology has a strong development, applied in many areas of life and efficiency brought about is unable to support, greatly reducing the ineffectiveness of manual work less effective accuracy of Man before Electronic computers are becoming more and more popular and accessible to everyone, and access to Information Management Systems is a top priority for organizations and businesses to manage their information. In the production and business activities, e-commerce has now confirmed the role of promoting and promoting the development of enterprises in the production as well as the management. For a hotel information management system in the hotel management is built to address the management needs of the hotel. The system will help businesses handle the job accurately, quickly, cost savings, human resources, but also promote the image of the hotel to friends inside and outside the province, in the country and internationally! . Interoperability, joint ventures with other Hotel Systems in the same field of activity is very high. The Management Information System will overcome the weaknesses of existing hotel management, with the ability to upgrade and expand many functions in the future. The topic is a practical requirement in the management of hotels currently operating, due to the increasing demand for tourist numbers, service numbers, commercial competition, cost savings The operation and, more importantly, the efficient and effective operation of the new management is the key to the success of our business. It is the most effective way to test the knowledge you have already learned and is the most effective way to approach it.

## 1.2. The Current System

Through our hotel survey we can see that the hotel is in direct form. With the current system customers are not provided information quickly, conveniently. When booking, guests will have to go to the hotel directly to book and pay. This will cost customers time and travel costs, thus reducing business efficiency. The promotion of hotels through the mass media. However, this type of hotel promotion is quite cost effective yet not really high, because the duration of advertising is limited so the introduction of the hotel in full detail is not possible. With the above limitations of the system certainly the hotel will not be able to achieve maximum business efficiency.

## 1.3. The Proposed System

The system has rights to use for guest, member, and admin.

Web applications need the following features, such as registering candidates.

Registration fee:

|  |  |
| --- | --- |
| Monthly | 50$ |
| Yearly | 500$ |

While registering candidates, some validation needs to be maintained as if the age of the applicant under 18 years of age is not allowed to make the registration.

Website display features should be divided into two categories:

• Unregistered users

• Registered users

It should contain the links:

1. **Home Page:** It should contain all the links where the user can navigate. It should also contain the section where the registered user can Login.
2. **Registration:** Here the user will be allowed to create their account. It should ask the details like:
   * Account Details
     + User Name
     + Password
   * Basic Details
     + First Name
     + Middle Name
     + Last Name
   * Date of Birth
     + Contact Number
3. **Search Page:** Here it should have the option of Searching the Match. Searching should be either based on:
   * Search By Room
   * Search By User Id

Here the profile details of the searched candidate should come without any contact details.

1. **Services:** Here the advertisement can be done on the newspaper classified. It should have the following fields—
   * Select Edition (selecting the particular state or country)
   * Payment
2. **Contact Us Page:** Here the Help line address and Contact details must be stated.

**Registered User:** All the features of above will be there. It should contain some other links as:

Home page of the registered user must have the section where related matches candidate to their profile must be stated. They candidate must be presented with their image and contact details and if the image is clicked then the profile should come on the display where apply button must be there to sent the request.

## 1.4. Boundaries of the System

Due to time limitations should eProject program have some limitations

## 1.5. Hardware and Software Requirements

### 1.5.1. Minimum Requirements

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| Hardware | \* Ram 1GB above  \* CPU 3.0 Ghz above  \* HDD 500 MB Drum upward  \* Connect to the Internet and LAN | \* Ram 512 MB above  \* CPU 2.2 Ghz above  \* Internet connection |
| Software | \* Windows Server 2003 Upward or Linux server.  \* Java | \* Web Browser |

### 1.5.2. Recommended Requirements

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| Hardware | \* Ram 2 GB Or more  \* CPU 3.0 Ghz above  \* HDD 5 GB Drum upward  \* Connect to the Internet and LAN | \* Ram 1 GB above  \* CPU 2.4 Ghz above  \* Internet connection |
| Software | \* Windows Server 2003 Upward or Linux server.  \* Java | \* Web Browser |

# 2. Customer Requirements Specification

## 2.1. Users of the System

* Admin
* User

## 2.2. System functions

### 2.2.1. User

**- Customer:**

* Login:

|  |  |
| --- | --- |
| **Description** | Allow login into User page |
| **Input** | Username and Password |
| **Process** | Check Username and Password on Database |
| **Output** | If don’t success, display error message. Opposite, allow access Member page with User acount. |

* Add new User

|  |  |
| --- | --- |
| **Description** | Allow admin to add new User. |
| **Input** | Input information of User. |
| **Process** | Check information accept information above on Database Update into Database. |
| **Output** | If don’t success, display error message. Opposite, display information of User has updated. |

* Edit/Update User

|  |  |
| --- | --- |
| **Description** | Allow admin to edit or update User. |
| **Input** | Input information of Member want to edit or update. |
| **Process** | Check information accept information above on Database Update into Database. |
| **Output** | If don’t success, display error message. Opposite, display information of User has edit/updated. |

* Search User:

|  |  |
| --- | --- |
| **Description** | Allow admin to search User by attributes(Name,….. ) |
| **Input** | Input info of User want to search. |
| **Process** | Find information above on Database. |
| **Output** | If don’t success, display error message. Opposite, display information of Member has found. |

* Update User:

|  |  |
| --- | --- |
| **Description** | Allow admin to update User |
| **Input** | Input information of User want to update |
| **Process** | Update into Database. |
| **Output** | If don’t success, display error message. Opposite, display message and information of User has updated. |

* Delete User:

|  |  |
| --- | --- |
| **Description** | Allow admin to delete User. |
| **Input** | Choose User want to delete. |
| **Process** | Delete all of User information on Database. |
| **Output** | If don’t success, display error message. Opposite, display information of Orders has found. |

### 2.2.3. Admin

* Admin account login:

|  |  |
| --- | --- |
| **Description** | Allow login into Admin page |
| **Input** | Username and Password |
| **Process** | Check Username and Password on Database |
| **Output** | If don’t success, display error message. Opposite, allow access into Admin page. |

* Log off the administrator account:

|  |  |
| --- | --- |
| **Description** | Sign out of the admin account of the app |
| **Input** | Click the log out button on the application's admin page |
| **Process** | Cancels the entire session of the current administrator. |
| **Output** | Go back to the login page |

* Change admin password:

|  |  |
| --- | --- |
| **Description** | Administrators after logging in to the system may change their passwords. |
| **Input** | Old password and new password. |
| **Process** | - Check whether the input is valid.  - Connect to the database and update the changed password. |
| **Output** | - Go to the login page if the password update is successful. |

* Search for members:

|  |  |
| --- | --- |
| **Description** | Administrators can search for members in a thread in the Username. |
| **Input** | Enter the string to look up in the member's Username. |
| **Process** | Search the database. |
| **Output** | Show all members have Username satisfied. |

* Manage members:

|  |  |
| --- | --- |
| **Description** | Administrators have the right to view information, edit or delete member accounts. |
| **Input** | Select the corresponding command of the account to be manipulated. |
| **Process** | - Function view customer information: display full customer information but not allow edit information.  - Account Deletion: Deleted accounts are not allowed in the app  - Edit account: Edit member account information |
| **Output** | - Full information of customers.  - A notification of success or error deleting member account. |

* Room Manager:

|  |  |
| --- | --- |
| **Description** | Administrators can:  - Add, edit, delete rooms in the database. |
| **Input** | Select and use pre-designed tools for product management and maker. |
| **Process** | - Add: Check the properties of the new room and add to the database.  - Edit: allows editing of room information  - Delete: if the room is not allowed to delete. If the room has been set, it is not allowed to delete. |
| **Output** | Show a success notification or an error message. |

* Manage room rates:

|  |  |
| --- | --- |
| **Description** | Administrators can:  - Add, edit, delete room rates in the database. |
| **Input** | Select and use pre-designed tools for product management and maker. |
| **Process** | - Add: Check the properties of the new room and add to the database.  - Edit: allows editing of room information  - Delete: if the room if the room is not priced. Not if the price has been set |
| **Output** | Show a success notification or an error message. |

* Unit Management:

|  |  |
| --- | --- |
| **Description** | Administrators can:  - Add, edit, delete units in the database. |
| **Input** | Select and use pre-designed tools for product management and maker. |
| **Process** | Add: Check units added to the database.  - Edit: allows editing of room information  - Delete: delete the selected number. |
| **Output** | Show a success notification or an error message. |

* Customer Management:

|  |  |
| --- | --- |
| **Description** | Administrators can:  - Add, edit, delete customer in the database. |
| **Input** | Select and use pre-designed tools for product management and maker. |
| **Process** | Add: Check customer added to the database.  - Edit: allows editing of room information  - Delete: delete the selected number. |
| **Output** | Show a success notification or an error message. |

* Management room type:

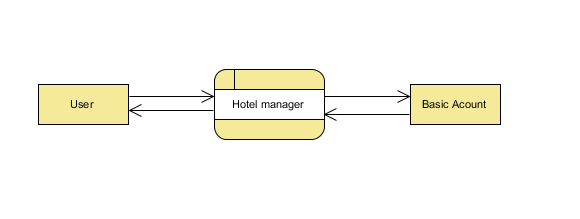
|  |  |
| --- | --- |
| **Description** | Administrators can:  - Add, edit, delete room type in the database. |
| **Input** | Select and use pre-designed tools for product management and maker. |
| **Process** | Add: Check customer added to the database.  - Edit: allows editing of room type information  - Delete: delete the selected number. |
| **Output** | Show a success notification or an error message. |

* Management of goods and services :

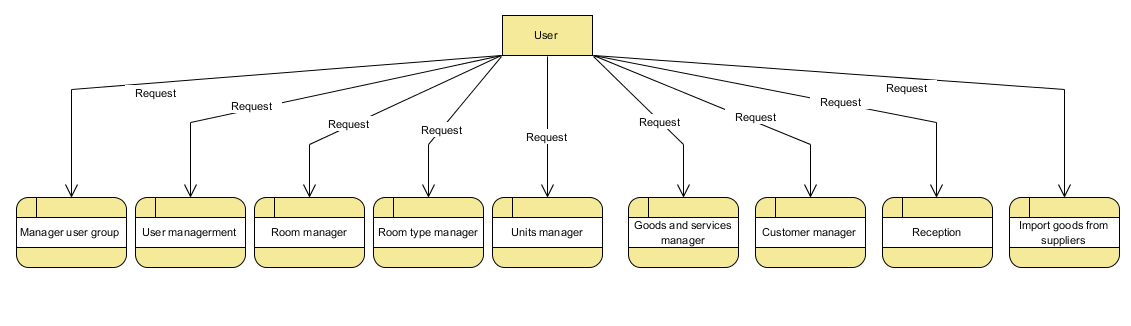
|  |  |
| --- | --- |
| **Description** | Administrators can:  - Add, edit, delete Price detail in the database. |
| **Input** | Select and use pre-designed tools for product management and maker. |
| **Process** | Add: Check customer added to the database.  - Edit: allows editing of Price detail information  - Delete: delete the selected number. |
| **Output** | Show a success notification or an error message. |

# 3.System Designs

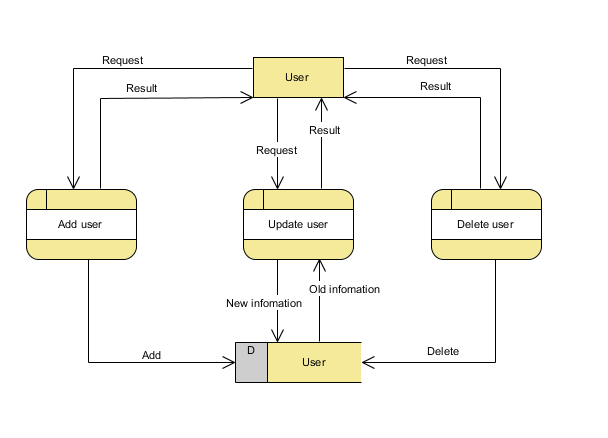
## Architecture



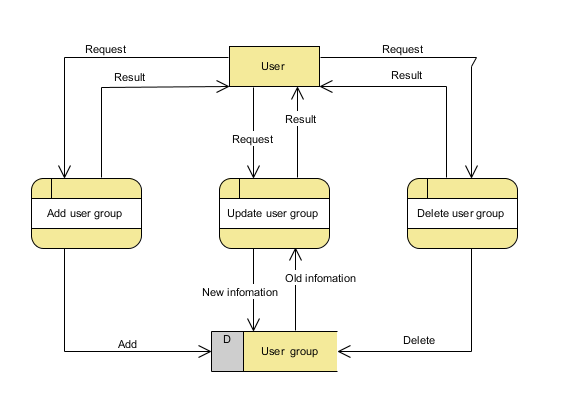
## 3.2. Data Flow Data (DFD)

****

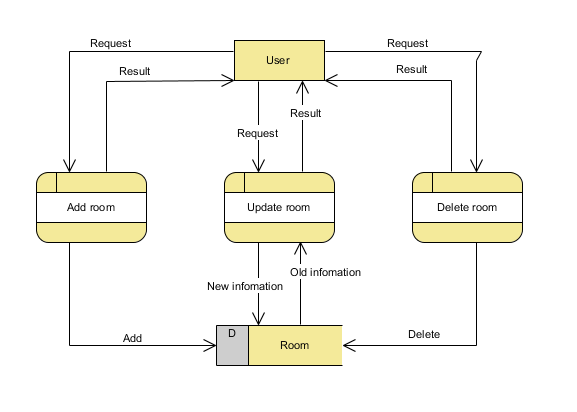
DFD Level 0: Context diagram.



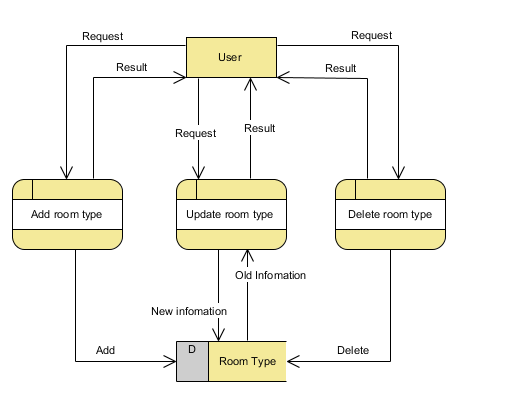
DFD : User management



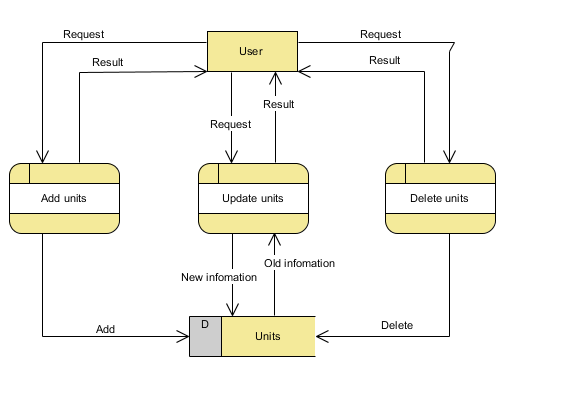
DFD : Manager user groups



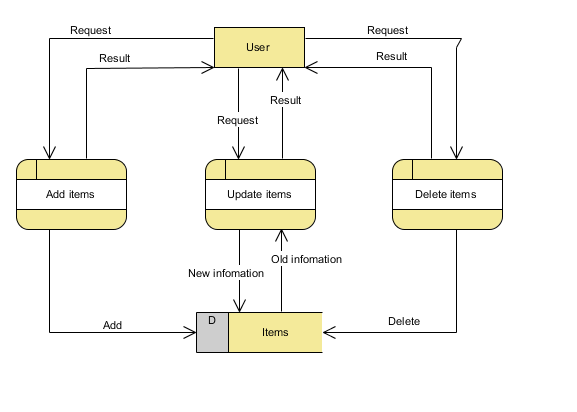
DFD : Room Manager



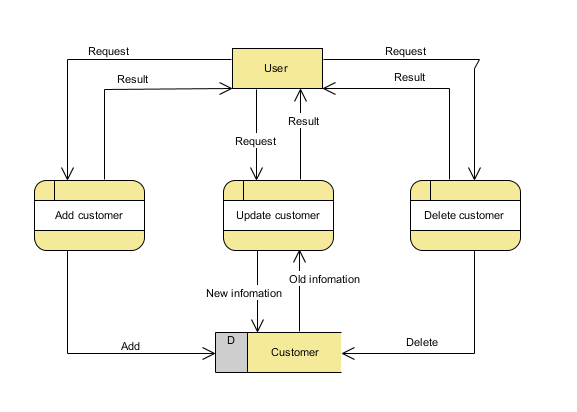
DFD : Room type management.



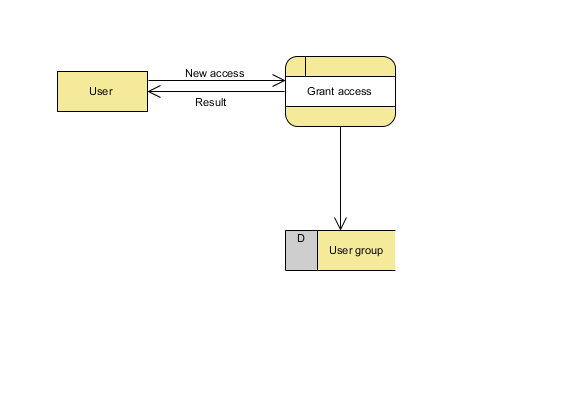
DFD: Units management.



DFD: Manage goods and services.



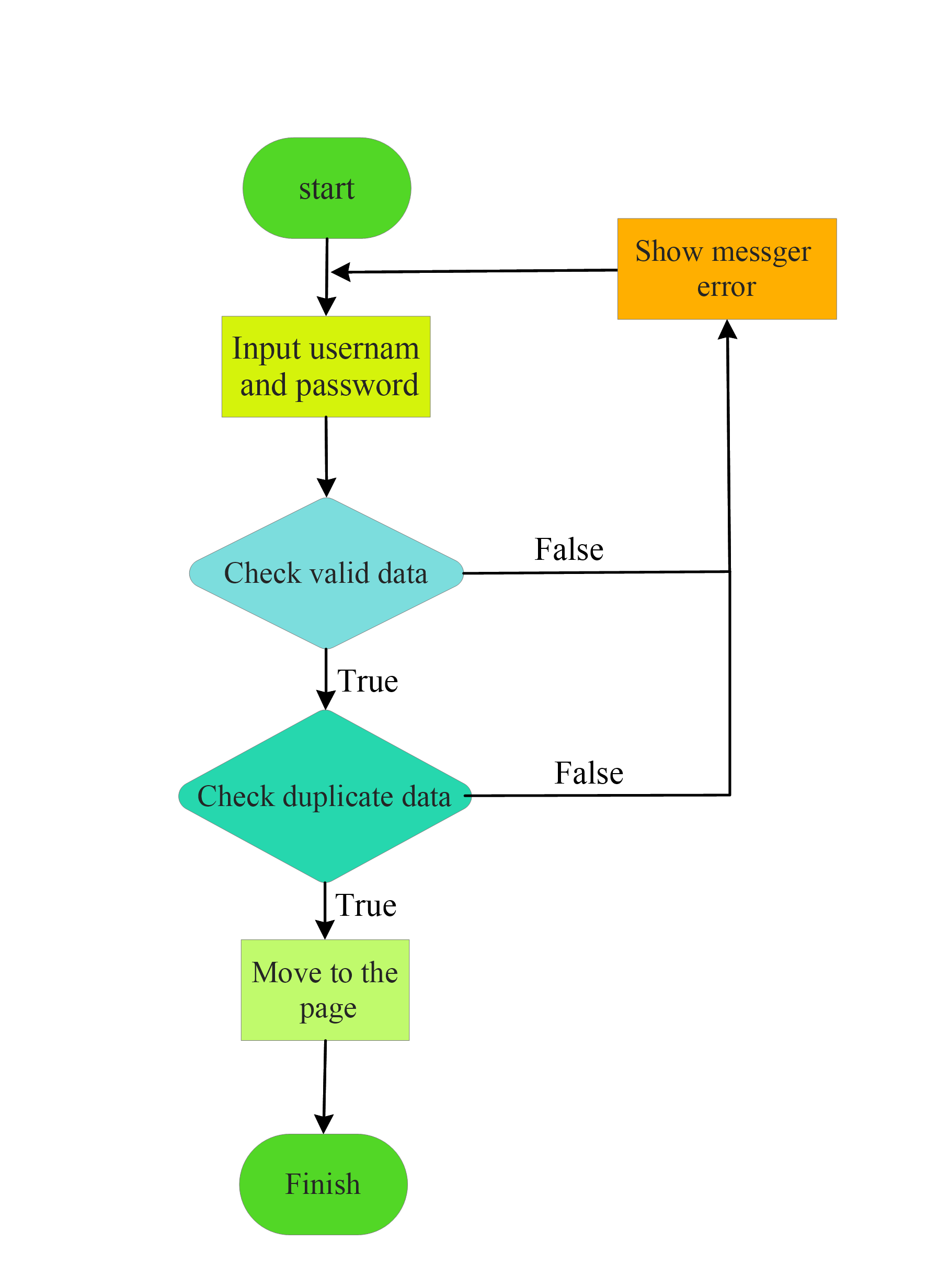
DFD: Customer Managerment.



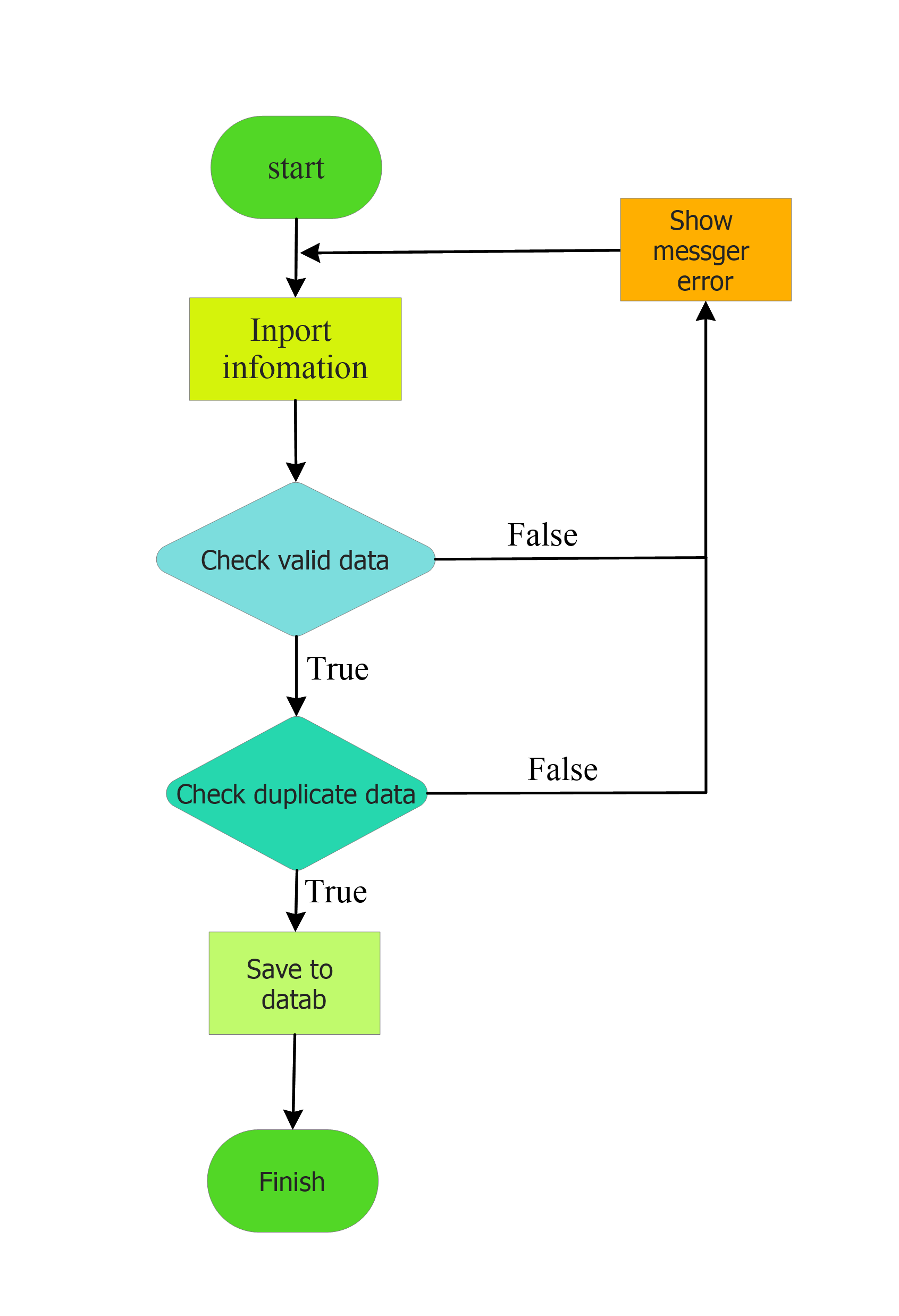
DFD: Grant access.

## Flow Chart

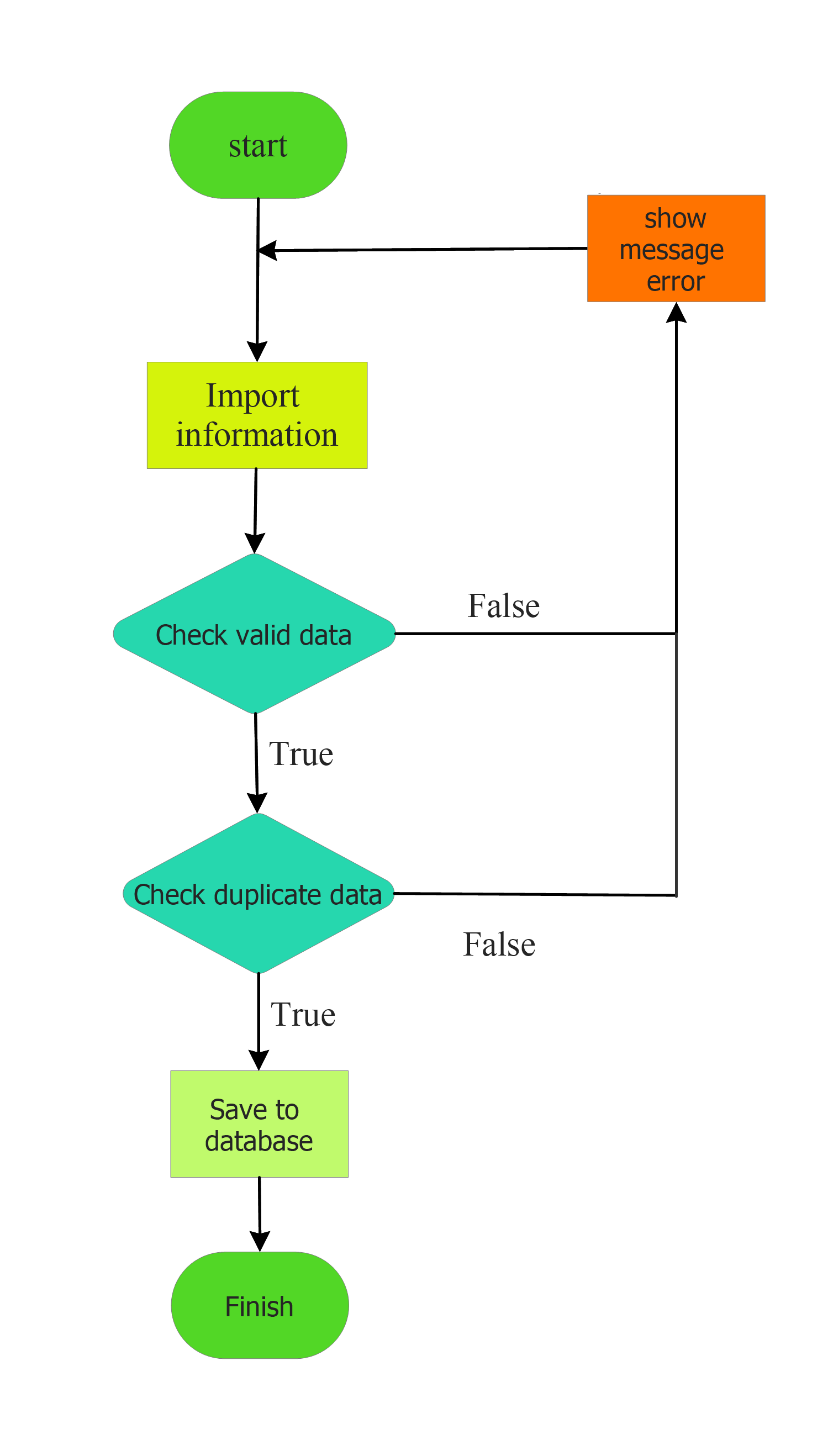
* Login:

****

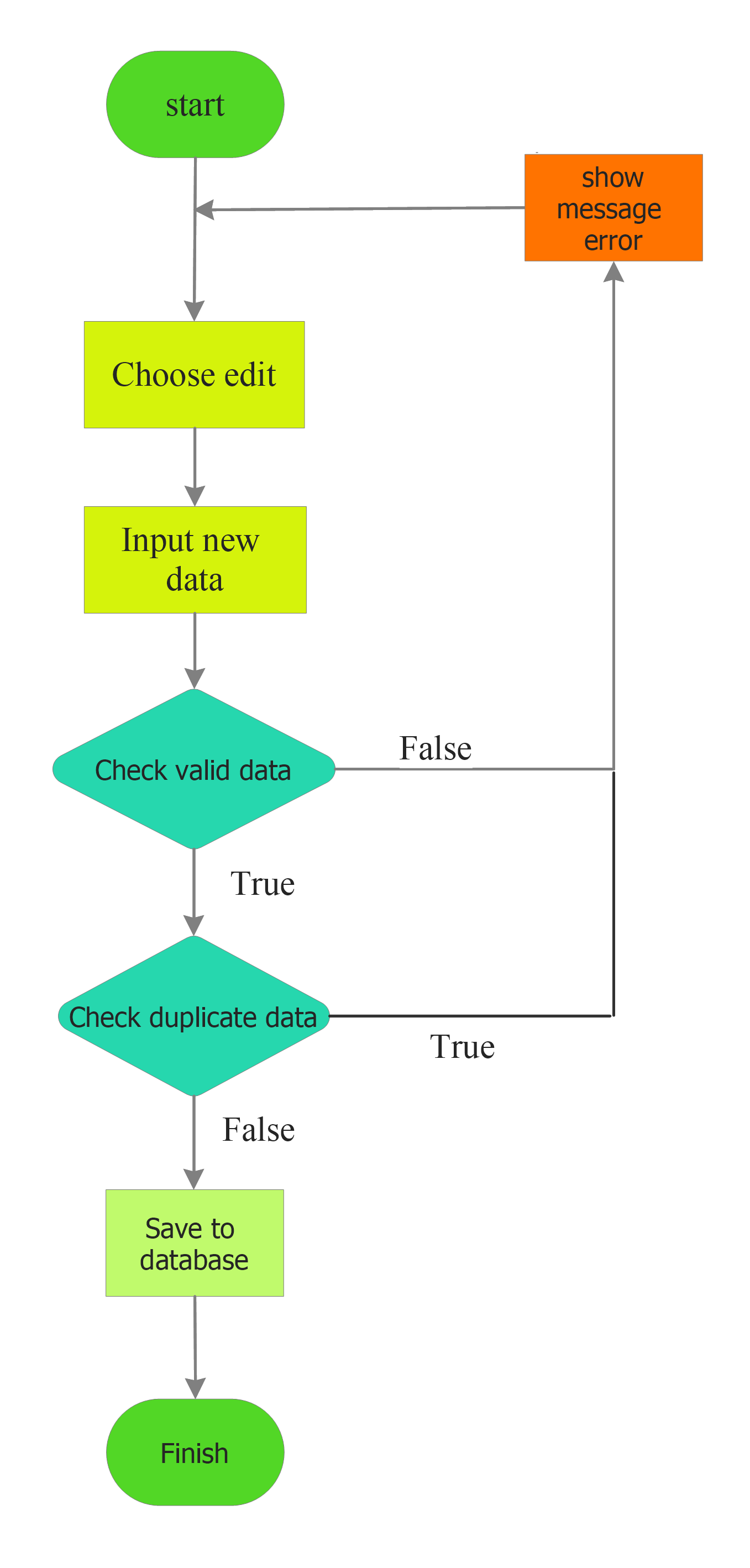
* Register



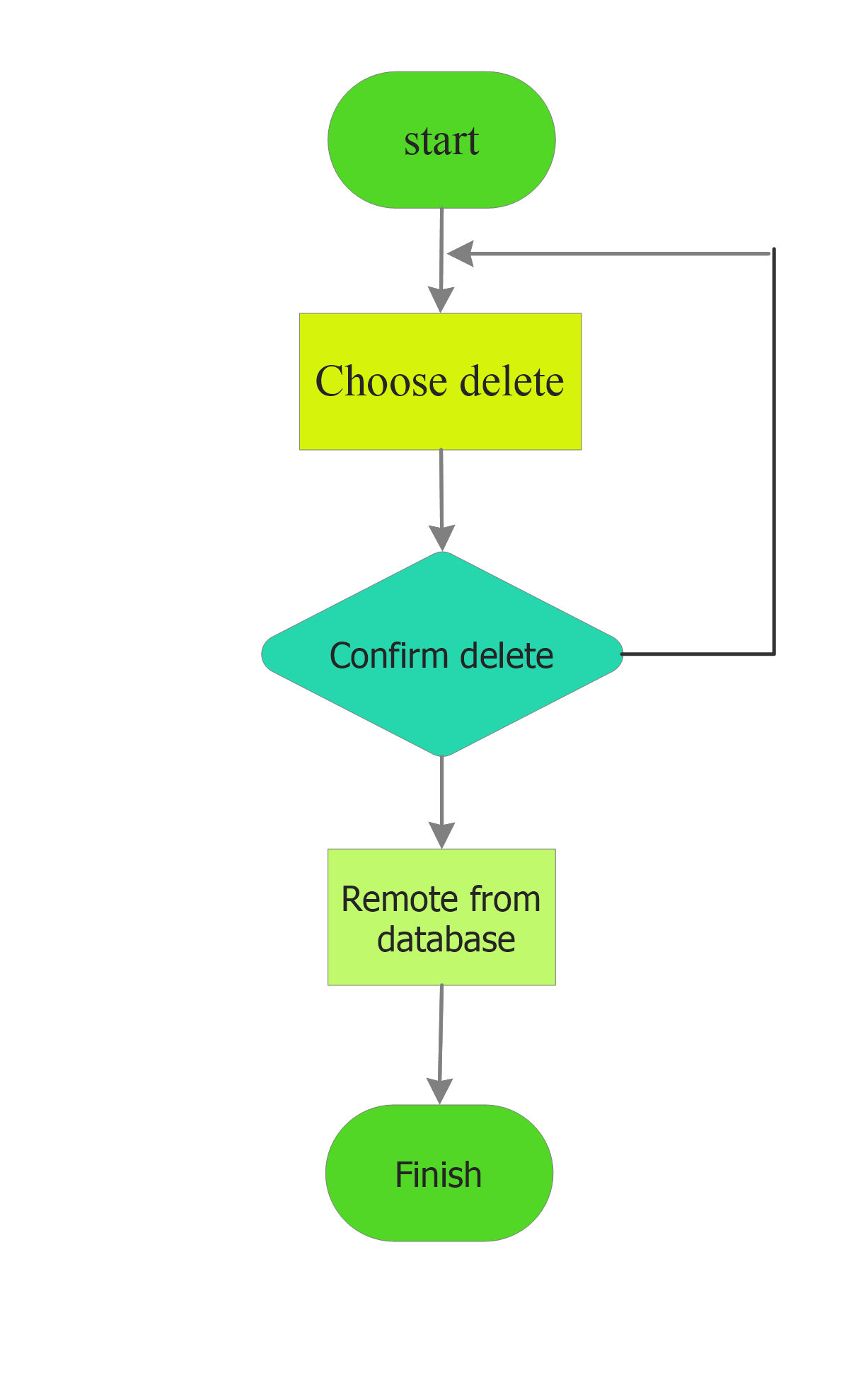
* Add:



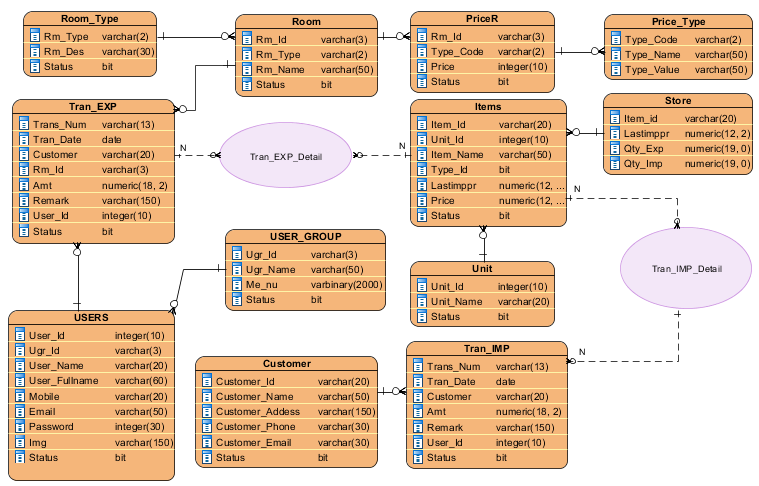
* Edit :



* Delete:



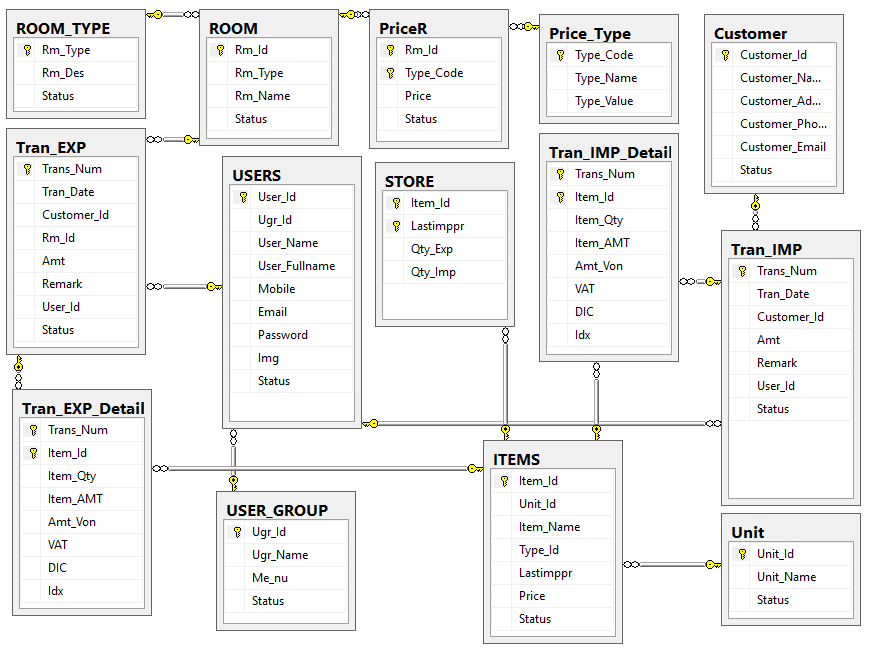
## Entity Relationship Diagram



## Database Structure

| ***No.*** | ***Table name*** | ***Description*** | ***Reference to*** |
| --- | --- | --- | --- |
|  | USERS | Used to store all administrator credentials | Admin, User |
|  | USER\_GROUP | Used to store all user information | Admin, User |
|  | PriceR | Used to store information of room rates | Admin, User |
|  | Price\_Type | Used to store room type information | Admin, User |
|  | Customer | Used to store customer information | Admin, User |
|  | ITEMS | Used to store information of goods | Admin, User |
|  | Unit | Used to store the amount of information | Admin, User |
|  | STORE | Used to store inventory | Admin, User |
|  | Tran\_IMP\_Detail | Used to store the import details | Admin, User |
|  | ROOM | Used to store the room. | Admin, User |
|  | ROOM\_TYPE | Used to store room type. | Admin, User |
|  | Tran\_EXP\_Detail | Used to store item details | Admin, User |
|  | Tran\_EXP | Used for storing Exported Goods | Admin, User |
|  | Tran\_IMP | Used to store imported goods | Admin, User |

### 3.5.1. Database Diagram



### 3.5.2 Table of database

**Table 1: USER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : USER** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **User\_Id** | int | No | PK | Store the user code of user |
| Ugr\_Id | Varchar(3) | No |  | Store the user group of user |
| User\_Name | Varchar(20) | No |  | Store the username of user |
| User\_Fullname | nvarchar(60) | No |  | Store the full name of user |
| Mobile | varchar(20) | No |  | Store the phone of user |
| Email | nvarchar(50) | No |  | Store the email of user |
| Password | varchar(30) | No |  | Store the pass of user |
| Img | nvarchar(150) | No |  | Store the image of user |
| Status | bit | No |  | Store the status of user |

**Table 2: USER\_GROUP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : USER\_GROUP** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Ugr\_Id** | int | No | PK | Store the User group code of user |
| Ugr\_Name | Varchar(3) | No |  | Store the user group name of user group |
| Me\_nu | Varchar(20) | No |  | Store the menu of user group |
| Status | bit | No |  | Store the status of user group |

**Table 3: PriceR**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : PriceR** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Rm\_Id** | Varchar(3) | No | PK | Store the rpriceoom code of pricer |
| **Type\_Code** | Varchar(2) | No | PK | Store the Price type of pricer |
| Price | numeric(18, 0) | No |  | Store the price of pricer |
| Status | bit | No |  | Store the status of pricer |

**Table 4: Price\_Type**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : Price\_type** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Type\_Code** | Varchar(2) | No | PK | Store the type code of Price\_type |
| Type\_Name | nvarchar(50) | No |  | Store the name of Price\_type |
| Type\_Value | varchar(50) | No |  | Store the value of Price\_type |

**Table 5: Customer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : Price\_type** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Customer\_Id** | varchar(20) | No | PK | Store the id of Customer |
| Customer\_Name | nvarchar(50) | No |  | Store the name of Customer |
| Customer\_Address | nvarchar(150) | No |  | Store the address of Customer |
| Customer\_Phone | nvarchar(30) | No |  | Store the phone of Customer |
| Customer\_Email | nvarchar(30) | No |  | Store the email of Customer |
| Status | bit | No |  | Store the status of Customer |

**Table 6: ITEMS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : Price\_type** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Item\_Id** | varchar(20) | No | PK | Store the commodity codes of ITEMS |
| Unit\_Id | int | No |  | Store the Code number of ITEMS |
| Item\_Name | nvarchar(50) | No |  | Store the goods name of ITEMS |
| Type\_Id | bit | No |  | Store the Commodity code of ITEMS |
| Lastimppr | numeric(12, 2) | No |  | Store the Price input price of ITEMS |
| Price | numeric(12, 2) |  |  | Store the price of ITEMS |
| Status | bit | No |  | Store the status of ITEMS |

**Table 7: Unit**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : Units** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Unit\_Id** | varchar(20) | No | PK | Store the id of Unit |
| Unit\_Name | int | No |  | Store the name unit of Units |
| Status | bit | No |  | Store the status of Units |

**Table 8:** **Tran\_IMP\_Detail**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : Tran\_IMP\_Detail** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Trans\_Num** | varchar(13) | No | PK | Store the transaction number of Tran\_IMP\_Detail |
| **Item\_Id** | varchar(20) | No | PK | Store the commodity codes of Tran\_IMP\_Detail |
| Item\_Qty | numeric(10, 2) | No |  | Store the Quantity of goods of TRANS\_IMP |
| Item\_AMT | numeric(10, 2) | No |  | Store the Export value of Tran\_IMP\_Detail |
| Amt\_Von | numeric(12, 2) | No |  | Store the Input value of Tran\_IMP\_Detail |
| VAT | numeric(12, 2) | No |  | Store the Percent value added of Tran\_IMP\_Detail |
| DIC | numeric(10, 2) | No |  | Store the discount of Tran\_IMP\_Detail |
| Idx | iint | No |  | Store the Index position in invoice list of Tran\_IMP\_Detail |

**Table 9:** **Tran\_EXP\_Detail**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : Tran\_EXP\_Detail** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Trans\_Num** | varchar(13) | No | PK | Store the transaction number of Tran\_EXP\_Detail |
| **Item\_Id** | varchar(20) | No | PK | Store the it unit of Tran\_EXP\_Detail |
| Item\_Qty | numeric(10, 2) | No |  | Store the Quantity of goods of Tran\_EXP\_Detail |
| Item\_AMT | numeric(10, 2) | No |  | Store the Export value of Tran\_EXP\_Detail |
| Amt\_Von | numeric(12, 2) | No |  | Store the Input value of Tran\_EXP\_Detail |
| VAT | numeric(12, 2) | No |  | Store the Percent value added of Tran\_EXP\_Detail |
| DIC | numeric(10, 2) | No |  | Store the discount of Tran\_EXP\_Detail |
| Idx | int | No |  | Store the Index position in invoice list of Tran\_EXP\_Detail |

**Table 10: ROOM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : ROOM** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Rm\_Id** | varchar(3) | No | PK | Store the id of Room |
| Rm\_Name | nvarchar(50) | No |  | Store the name of Room |
| Status | bit | No |  | Store the status of Room |

**Table 11: ROOM\_TYPE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : ROOM** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Rm\_Type** | varchar(2) | No | PK | Store the id of Room\_type |
| Rm\_Des | nvarchar(30) | No |  | Store the describe of Room\_type |
| Status | bit | No |  | Store the status of Room\_type |

**Table 12: Tran\_IMP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : Tran\_IMP** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Trans\_Num** | varchar(13) | No | PK | Store the transaction number of TRANSHEAD\_IMP |
| Begin\_Date | datetime | No |  | Store the begin date of TRANSHEAD\_IMP |
| End\_Date | datetime | No |  | Store the end date of TRANSHEAD\_IMP |
| Customer\_Id | varchar(20) | No |  | Store the id customer of TRANSHEAD\_IMP |
| Amt | numeric(12, 2) | No |  | Store the Total value of TRANSHEAD\_IMP |
| Remark | nvarchar(150) | No |  | Store the note of TRANSHEAD\_IMP |
| User\_Id | int | No |  | Store the id user of TRANSHEAD\_IMP |
| Status | char(1) |  |  | Store the status of TRANSHEAD\_IMP |

**Table 13: Tran\_EXP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : Tran\_EXP** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Trans\_Num** | varchar(13) | No | PK | Store the transaction number of TRANSHEAD\_EXP |
| Begin\_Date | datetime | No |  | Store the begin date of TRANSHEAD\_EXP |
| End\_Date | datetime | No |  | Store the end date of TRANSHEAD\_EXP |
| Customer\_Id | varchar(20) | No |  | Store the id customer of TRANSHEAD\_EXP |
| Amt | numeric(12, 2) | No |  | Store the Total value of TRANSHEAD\_EXP |
| Remark | nvarchar(150) | No |  | Store the note of TRANSHEAD\_EXP |
| User\_Id | int | No |  | Store the id user of TRANSHEAD\_EXP |
| Status | char(1) |  |  | Store the status of TRANSHEAD\_EXP |

**Table 14: STORE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table : STORE** | | | | |
| **Field name** | **Data type** | **Null** | **Key** | **Description** |
| **Item\_Id** | varchar(20) | No | PK | Store the goods code of Items |
| Lastimppr | numeric(12, 2) | No |  | Store the the Price input price of Items |
| Qty\_Exp | Int | No |  | Store the Quantity exported of Store |
| Qty\_Imp | int | No |  | Store the number of import of Store |

## 3.6 Screen Design

# *Checklists*

## Check List of Validation

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Option** | **Validated** |
| 1. | Can a new user who gets registered, enter the application after logging in? | Y |
| 2. | Do all the options present in the application display the correct result? | Y |
| 3. | Does the application’s functionality resolve the user problem, and satisfy their needs? | Y |
| 4. | Has the hardware and software been correctly chosen? | Y |
| 5. | Ensure data in constraint | Y |

## Submission Checklist

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Option** | **Yes** | **No** | **NA** | **Comments** |
| 1. | Are the users able to log in to the application after validation is performed on the user name and password? | Y |  |  |  |
| 2. | Are the users able to modify their details after getting registered? | Y |  |  |  |
| 3. | Are all the screen contents devoid of spelling mistakes? | Y |  |  |  |
| 4. | Is the user able to trade on fund which has been selected? | Y |  |  |  |
| 5. | Is the application user-friendly? | Y |  |  |  |