**Class Diagram**

There are many classes classified and attributes in the class diagram:

* **User**

User class has some attributes like Name, Email, Password, Contact and all are of String type.

* Name
* Email Id
* PasswordS

It also has some Functions like:

* getName()
* getEmail()
* bookMeeting()
* getPassword
* setPassword
* **roomOwner**

roomOwner class has a attribute:

* PropertyID()

It also has a Function called:

* addRoomDetail()
* **room**

Room class has some attributes :

* Location
* Price
* Status
* Rent
* Seller
* PropertyID

It also has some functions like:

* getRoomDetails()
* getStatus()
* setStatus()
* addRoomDetails()
* **List**

List class has a attribute :

* criteria

It also has some functions like:

* updateList()
* getPropertyDetails()
* displayList()
* **buyer**

buyer class has a attribute :

* Registered

It also has some functions like:

* checkList()
* bookRoom()
* viewRoomDetails()
* **payment**

payment class has a attribute :

* amount

It also has some functions like:

* getPrice()
* displayPaymentOptions()

**Case studies of testing Tool**

**Introduction**

Tools are great…except when they are not Gee-whiz tools often have gee-whiz price tags to go with them Okay, free tools are great…except when they are not great Tools, free or pricey, can waste your time if you misuse them As a tester and/or developer, you should know your options so that you can use them as a manager , you should know your team’s options so you can help to choose the right ones Let’s look at some examples of free tools use- successful Or unsuccessful.

**GUI Test Automation**

There are open-source tools for GUI test Automation Selenium is the one most commonly.

Mentioned Keyword-driven architecture is essential For maintainability Otherwise, you'll hit the same problems. That occurs with commercial tools Using an experienced lead is necessary for Good architecture.

**Here is the list of top Android application testing tools:**

* Robotium.
* SeeTest Continuous **Testing** Platform.
* MonkeyRunner.
* Ranorex.
* Appium.
* UI Automator.

**Unit Testing**

Most of our clients using continuous integration Extend it with unit testing. The unit family of tools is most typically used (Cpp-Unit, J-Unit, etc.). Some report developers using it for TDD. Code coverage analysis tools (e.g., gcov) are often used in conjunction with these tools. Many developers don't have training in proper test design, so these tests are less useful than they could be. Testers should learn the basics of unit testing, white-box test design, etc., and help developers get more value.

**Scripting Tools**

Many testers automate tests using scripting tools Ruby, Python, Tcl/Tk, and Unix shells are in common use If anything, there are too many different options, leading to Tower of Babel problem Scripting is best done by people with some programming skill and knowledge. However, testers can learn or even teach themselves Care must be taken to avoid maintainability. Problems Very large, complex, sophisticated test systems can be built.