# Chapter 1

## Introduction

### Background of System

Hostel is the accommodation house for the students where they used to stay in sharing basis as well as individually. Besides this it accepts individual or group of individuals who want to stay for the short-term. It provides common area and communal facilities to all the individuals living under the same house. Due to of which the flow the people throughout the year is very large and it became tough task to track the record of the individuals. Staffs are also one of the important individuals of the hostel.

To track those all those information, hostel management system has been purpose. By the help of this system, Hostel owner can easily track the information of students, Visitors as well as of the staff. He/she can easily keep record of the previous students and make them ease to extract any information wherever they want it. By the help of this system, they can track down the information of student weather he/she is in hostel or stayed away, can track down the information of room, staff, and block.

### Overview of the System

For the development of hostel management system, different technology should be used. C#.Net has been used as the programming language whereas MySQL as a database. The application will be develop using Visual Studio.

### About System

Hostel management system is the desktop application which can be installed on single computer. It helps the hostel owner to track the information of the individuals staying in the system. They can be well informed about the status of the hostel, room of the hostel and their status in few clicks. They can easily track down the moving in and moving out time of the students as well as of the staff. Student’s movements, staff activities, room status, block description all these can be well organized.

### 1.4 Aims and Objective

In Last few year, the number of educational institute is increased in high rate especially in urban area. Because of which the number of hostels also increasing for the accommodation of those students moving towards the city for education or jobs. Here there are a lot of presser on the hostel owner who is owns the hostel. This software helps them to overcome that strain and can easily manage the hostel. It helps him in easy identification and management of the hostel. The aims and objectives of the application has been described below: -

**Aims**

* To help the hostel owner to track the information of students easily and correctly.
* To digitalized the hostel in this modern era with new facilities for the management of the hostel.
* To decrease the strain of the hostel running people from their head by automating the works which were doing manually previously.

**Objectives**

* To allocate the students to different room easily
* To the update the status of the student status in case if he/she left the hostel
* To track down the staff as well as student status
* To monitor the vacant room in hostel
* To edit the detail of the student and staff records with few clicks.
* To control the status of the fee Payment

# Chapter 2

## 2.1 Introduction

Before starting any coding, analysis is to be done about that system. Analysis is done on the existing system if there is any install in any of the hostel. By doing the analysis, client requirements can be understood and helps in gathering requirements which make the process of development easy and system can be developed as per client requirement. Problems with existing system can be identified so that it can be correct in new solution.

**Rich Pictures**

Rich picture is graphical representation of the situation to solve specific problem. It consists images, icon, texts, symbols so that it the problem can be solve graphically. It represents the main elements of the system and their relationship.

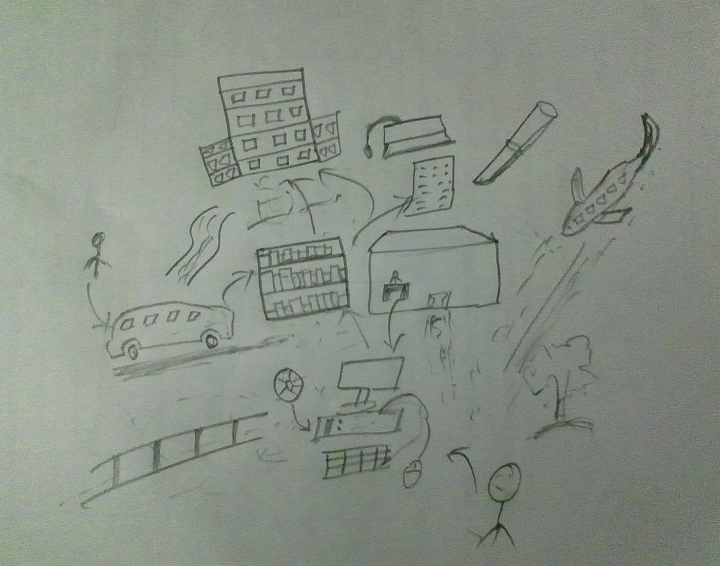


Fig: Rich Picture

## 2.2 Information Gathering Techniques

It can be defined as the techniques to gather information during the process of analysis. It can be done by applying different process to obtain data from the audience. Some of the techniques I have used has been discussed below which I apply during the process of analysis.

* **Interviews:** It is the process of data collection in which two or more than individuals sit beside each other and discussed about particular topic. By using this technique, I communicate with hostel owner and discussed on the topic problems faced by him in management of hostel in the absence of software. I also tried to find out the requirement of the software from their side.
* **Focus Group:** In this techniques specific group of people are focused to find out the solution on specific processes. I visited to several hostel and talk with the students and try to find out the problem related to hostel management system. There answer was recorded in my phone and later I listen to that audio two, three times and find out the data related to this application.
* **Questionnaires:** In this process of data collection, sample questions are prepared and try to find out the solution from mass point of view. I prepared some sample questions related to the hostel management system and convince students who accommodated by the hostel.
* **Social Media:** Social media is the latest form of data collection internationally. I posted the issues relation to application on several pages to get the viewer opinion. By reading their comments, interest data were collected and later analyzed.

## 2.3 Feasibility Study

It is process of exploring whether the project is effective after its completion. It is the degree of being easily or confidentially done. It describes the practically of the thing being done. Feasibility study is the examination of the ability of doing project successfully. The following techniques were used for the feasibility study and are explained below:

* **Operation Feasibility:** It is the study whether the existing problem can be solved by the introduction of new system or not. By doing operation feasibility we can take advantage of opportunities by solving the previous problem which has been facing previously. For this project I report some problem which has been faced from the existing system and some requirements was also collected during the process of operation feasibility.
* **Technical Feasibility:** It is the study in which technical portion of the project get involved like input, processes, and output. If we are planning to the long term solution then we must think of the technical feasibility study. By doing technical feasibility study, future problem can be tracked and can be concluded to solution as soon as possible.
* **Economic Feasibility:** It is the study whether the costing of the project is feasible and logical. By doing economic feasibility study, it can predict whether the project can be completed in estimated budget or not. In the process of project development, economic feasibility is calculated by analyzing the cost of development and revenue generated from business.
* **Social Feasibility:** It is the study of the social acceptance of the project. After the development of project, the society will accept it or not. All this comes under social feasibility study. It also describes the project effects on society after the development of it.

## 2.4 Analysis Methodology

The methodology I am undertaking for this project is agile methodology. The cages can be made into the project at any phase of the development. Agile methodology accepts the changes at any phase of software development. As investigating more and deeper, changes comes out in the project. To adopt those changes agile methodology will be the best match.

Agile methodology is iterative approach of developing software. In this methodology project is divided into small chunks into several part. And the work is to be done on each chunks. After completing every chunks, the later merge and output a complete solution. In the process of development, every chunks are tested well only then we are allowed to move into next chunk. Each and every chunks are to be tested individually.

Before starting the development of the project all the articles and papers were collected from analysis and were read properly. After that the project was discussed with our project manager and he suggest me to bring some changes in the project. So all those changes were made.

## 2.5 System Requirements Specifications

The system requirements specification is the identification of requirements in the process of development. This process is done before starting any coding. It is very important phase for the periodization of requirements.

### **Functional requirements**

In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs.

Functional requirements may involve calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases. Functional requirements are supported by non-functional requirements, which impose constraints on the design or implementations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Title** | **Description** | **Rational** | **Dependencies** |
| FR1 | User Registration | Hostel Owner has to register with the system | To use the system | N/A |
| FR2 | User Login | User has provide username and password to login into the system | To authorized user into the system | N/A |
| FR3 | Add Student | User has to input all the details of the system to add new student | To add student | FR18, FR13 |
| FR4 | Edit Student | User can select student or search student to edit student information | To edit student | FR3 |
| FR5 | Delete Student | User can delete student by selecting student Information | To delete student | FR3 |
| FR6 | Search Student | User can search student by entering student Id | To search student | FR3 |
| FR7 | View Student | User can view information of student by selecting or by searching | To view student | FR3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FR8 | Add Staff | User has to provide all the details of the staff to add new staff | To add new staff | FR13 |
| FR9 | Edit Staff | User can edit staff detail by selecting staff or searching staff | To edit staff | FR8 |
| FR10 | Search Staff | User can search staff by using staff Id | To search staff | FR8 |
| FR11 | Delete Staff | User can delete staff by selecting | To delete staff | FR8 |
| FR12 | View Staff | User can view staff by searching or b selecting | To view staff detail | FR8 |
| FR13 | Add Block | User can add block by providing block details | To add new block | N/A |
| FR14 | Edit Block | After edit block by selecting or searching | To edit block information | FR13 |
| FR15 | Delete Block | User can delete block by selecting or searching | To delete block information | FR13 |
| FR16 | View Block | User can view block by searching or selecting | To view bock | FR13 |
| FR17 | Search Block | User can search block by using block id | To search block | FR13 |
| FR18 | Add Room | User can add Room by providing room detail | To add new room | FR13 |
| FR19 | Edit Room | User can edit room by searching or selecting | To edit room information | FR18 |
| FR20 | Delete Room | User can delete room by selecting or searching | To delete room | FR18 |
| FR21 | Search Room | User can search room by using room id | To Search room | FR18 |
| FR22 | View Room | User can view room detail by selecting or searching | To view room detail | FR18 |
| FR23 | Add Course | User can add course by providing course detail | To add course | N/A |
| FR24 | Edit Course | User can edit course by selecting or searching | To edit Course | FR23 |
| FR25 | Search Course | User can search by using course id | To search course | FR23 |
| FR26 | Delete Course | User can delete course by using course id | To delete course | FR23 |
| FR27 | Logout | User can logout from the system | To logout from system | FR2 |

### **ii) Non-functional Requirements:**

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Title** | **Description** | **Rational** | **Dependencies** |
| NFR1 | Performance | Accomplishment of the useful works | To complete all the required task with accuracy | N/A |
| NFR2 | Security | Save the application from external threats as well as to secure the data of the system | To secure data from eternal entity | N/A |
| NFR3 | Availability | System is functioning well under normal conditions and meeting client requirements | Available all the time when there is it’s need | NFR1, NFR2 |
| NFR4 | Reliability | System will not fail during the time of operation | To developed trust with the system | NFR2 |
| NFR5 | Maintainability | If any types of errors occur into the system, it can be repaired and brings into previous state as soon as possible | To repair problem as soon as possible if an problems occur in the system | NFR3 /NFR4 |
| NFR6 | Portability | Application movement from one environment to another | To Adopt new environment easily | NFR3 |
| NFR7 | Usability | Maintaining quality of the product when working on it |  | NFR1/NFR2 |

**Prioritization: -**

Prioritization is defined as managing the requirements according to their urgency and relative importance. By the process of prioritization, the critical requirements are supposed to do in the beginning if the project is running out of the budget.

For the process of prioritization, I have used MosCow approach. Following are the reasons why I have used this approach

• It coordinates basic and manage spending plans.

• It urges individuals to reexamine on their prerequisites.

• It urges individuals to diagram and oversee better work.

**Moscow (Must, Should, Could, Won’t) prioritization: -**

**Must have: -** A highly demanded requirement of the system. Without these requirements application cannot be proceeding further.

**Should have: -** Requirements which fulfill the initial purpose of the system.

**Could have: -**

It is an attractive requirements of the system to make the system fully functional. In the absence of these, application can perform their basic work.

**Won’t have: -**

Requirements which doesn’t affect the operation of the system

|  |  |  |
| --- | --- | --- |
| ID | Functional Requirement | Moscow |
| FR1 | Registration | Must have |
| FR2 | user Login | Must have |
| FR3 | Add student | Must have |
| FR4 | Edit Student | Should have |
| FR5 | Delete Student | Should have |
| FR6 | Search Student | Could have |
| FR7 | View Student | Won’t have |
| FR8 | Add Staff | Must have |
| FR9 | Edit Staff | Should have |
| FR10 | Delete Staff | Should have |
| FR11 | Search Staff | Could have |
| FR12 | View Staff | Won’t have |
| FR 13 | Add Block | Must have |
| FR14 | Edit Block | Should have |
| FR15 | Delete Block | Should have |
| FR16 | Search Block | Could have |
| FR17 | View Block | Won’t have |
| FR18 | Add Room | Must have |
| FR19 | Edit Room | Should have |
| FR20 | Delete Room | Should have |
| FR21 | Search Room | Could have |
| FR22 | View Room | Won’t have |
| FR23 | Add Course | Must have |
| FR24 | Edit Course | Should have |
| FR25 | Delete Course | Should have |
| FR26 | Search Course | Could have |
| FR27 | View Course | Won’t have |
| FR28 | Logout | Must have |

|  |  |  |
| --- | --- | --- |
| ID | Non-functional Requirements | Moscow |
| NFR1 | Performance | Should have |
| NFR2 | Security | Must have |
| NFR3 | Availability | Could have |
| NFR4 | Reliability | Could have |
| NFR5 | Maintainability | Should have |
| NFR6 | portability | Should have |
| NFR7 | usability | Should have |

## 2.6 Architecture

System architecture is conceptual representation of the system how the components of the system works together to make system fully functional. It describes the behaviors and structure of the system as well as views. It describes and represents the system so that it supports about the structure and behavior of the system.

**System Architecture**

As system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A System architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages. I have used Model, View and Controller Architecture for the development of application.

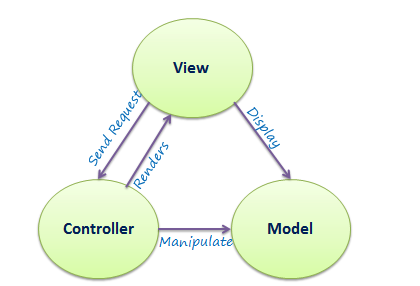


Fig: MVC Architecture

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

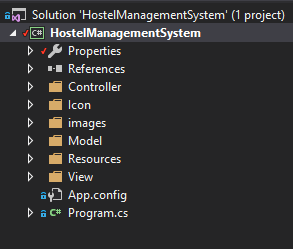
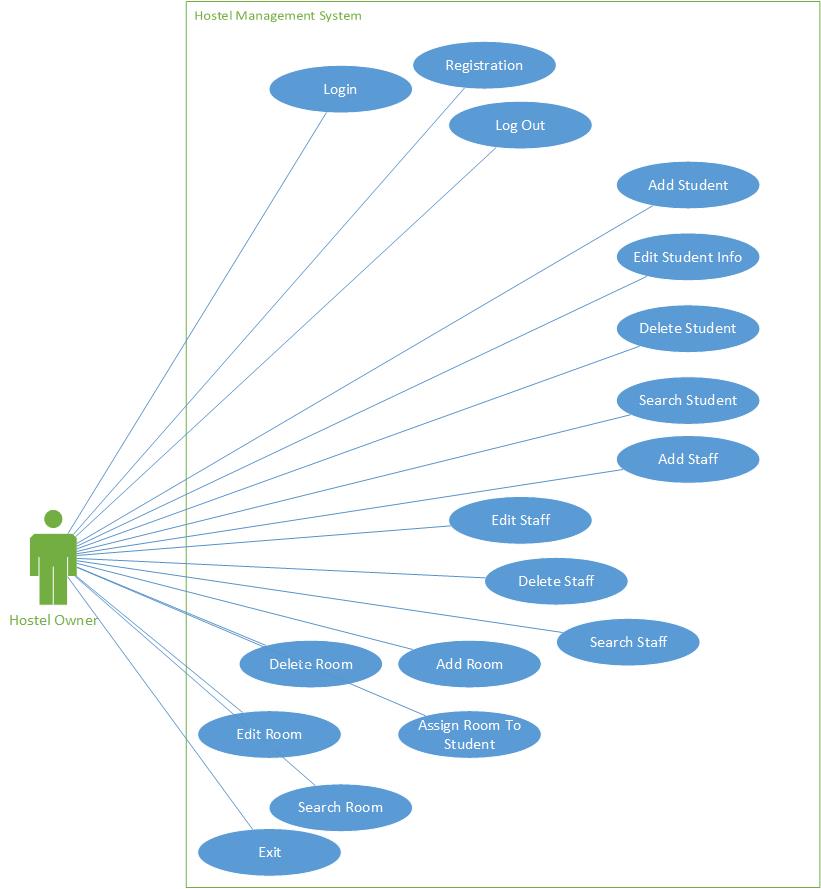


Fig: Screenshot of System Architecture of Hostel Management System

## 2.7 Use Case Diagram

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. In this case, Hostel management system is the system and User is the actor. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.



## 2.8 Natural language Analysis (NLA)

Natural language analysis is the process to find out the candidate classes. It helps in duplication of same classes to be created more than one time. It also helps to find out the relationship between entity and attributes of the entity. By doing Natural Language analysis the efficiency of the development can be increased and minimize the errors.

From the Natural Language Analysis following candidate’s classes were obtained:

|  |
| --- |
| Candidate Classes |
| System, Hostel, add, staff, Student, Block, College, Delete, Register, Course, Search, User, Room, Admin, Owner, Management, Login, Logout, Exit, View, Select, Customers, Visitors, Guest, Stay, Accommodation, Bed, Boys, Girls, Record, File, User , AddStudent, EditStudent, DeleteStudent, SearchStudent, AddStaff, EditStaff, DeleteStaff, SearchStaff, AddRoom, EditRoom, DeleteRoom, SearchRoom, AddCourse, EditCourse, DeleteCourse, SearchCourse, AddBlock, EditBlock, DeleteBlock, SearchBlock, ViewStudent, ViewCourse, ViewStaff, ViewBlock, ChangePassword |

From the above candidate classes and functionality were identified. All the reasonable noun words come under the category of Classes and Verb or Adverb comes under the category of Functionality. The table of Classes and Functionality is shown below:

|  |  |
| --- | --- |
| **Classes** | **Functionality** |
| Student, Staff, Course, Block, Room, User | Registration, Login, Exit, Logout, AddStudent, EditStudent, DeleteStudent, SearchStudent, AddStaff, EditStaff, DeleteStaff, SearchStaff, AddRoom, EditRoom, DeleteRoom, SearchRoom, AddCourse, EditCourse, DeleteCourse, SearchCourse, AddBlock, EditBlock, DeleteBlock, SearchBlock, ViewStudent, ViewCourse, ViewStaff, ViewBlock, ChangePassword |

## 2.9 Initial Class Diagram

Class Diagram is the graphical representation of classes and attributes. It represents the relationship between the classes. Datatype of attributes and function return type are also represented in class diagram. Single class is rectangular in structure and divided into three compartments. The three compartments are explained below:

* The uppermost compartment consists the name of class
* The middle compartment consists the name of attributes with their datatype
* The lowermost compartment consists the name of functions with their return type

There is some relationship which exist between classes. They are listed below with brief explanation:

* **Association:** When there needs to be communicate between two classes, association is used. As User directly communicate with student so there is association between them. It is represented in the diagram as below



Fig: Association between student and Instructor

* **Aggregation:**  When the class owns the child class object then the relationship between them is called as aggregation relationship. As for example the engine doesn’t exist in the absence of Car. From the below example it has been clear that Car consist of Engine and wheel which comes under the category of aggregation relationship.

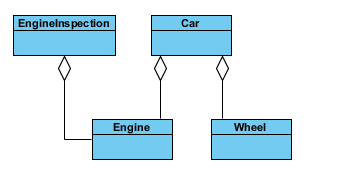


Fig: Aggregation

* **Composition:** When the two or more than two classes combine to form a single class and give them a meaning in that the relationship between them is Composition. In this case if Person is deleted then all head, Hand, Leg classes will have deleted.

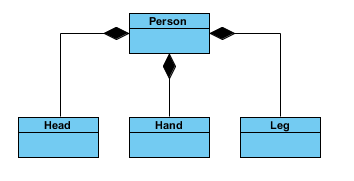


Fig: Image of Composition

* **Inheritance:** It is a type of relationship in which parent class obtain the property of child classes. If parent is deleted, then child class existence of itself.

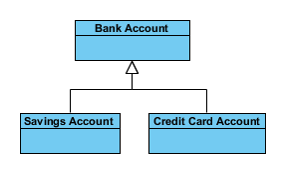


Fig: Inheritance

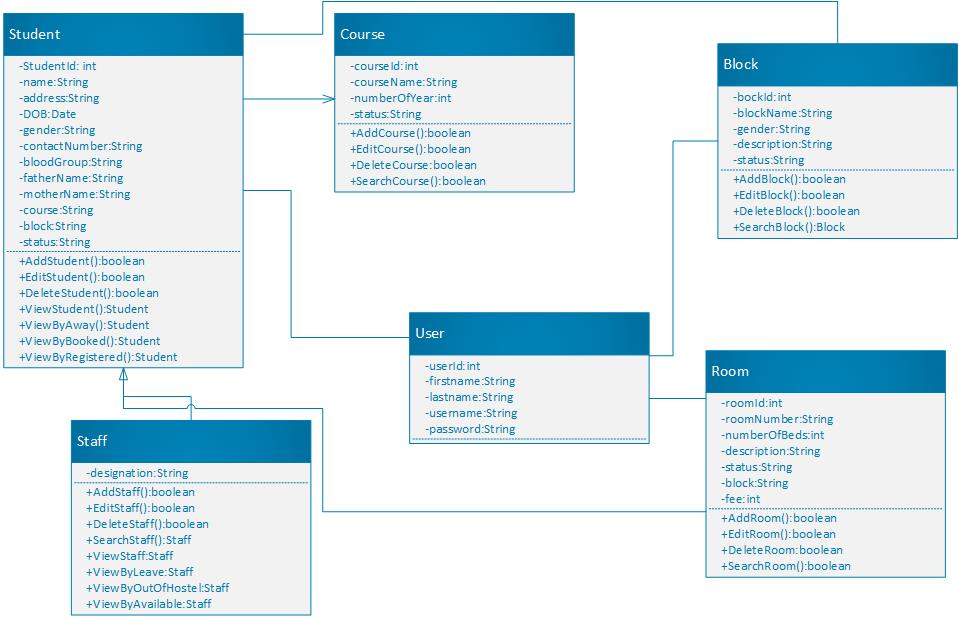


Fig: Class Diagram

# Chapter 3 Design

## 3.1 Introduction

System design is the process of defining the architecture, modules, interfaces and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

## 3.2 Tools

Tools which are used for the development of Hostel Management System are given below: -

* Integrated Development Environment(IDE) = Visual Studio
* Database = MySQL
* All the required diagram are made using Microsoft Visio

## 3.3 Structure Model

### 3.3.1 Class Diagram

Class Diagram is the graphical representation of classes and attributes. It represents the relationship between the classes. Datatype of attributes and function return type are also represented in class diagram. The final class diagram remain same as it was analyses in analysis section.

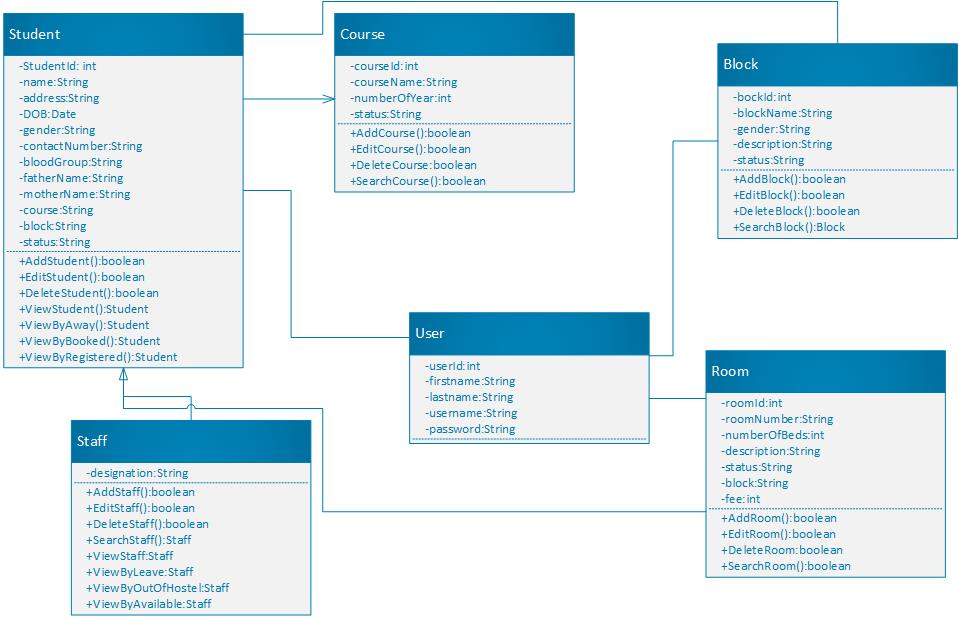


Fig: Class Diagram

### 3.3.2 ER Diagram

Entity relationship diagram is graphical representation of relationship between entities.

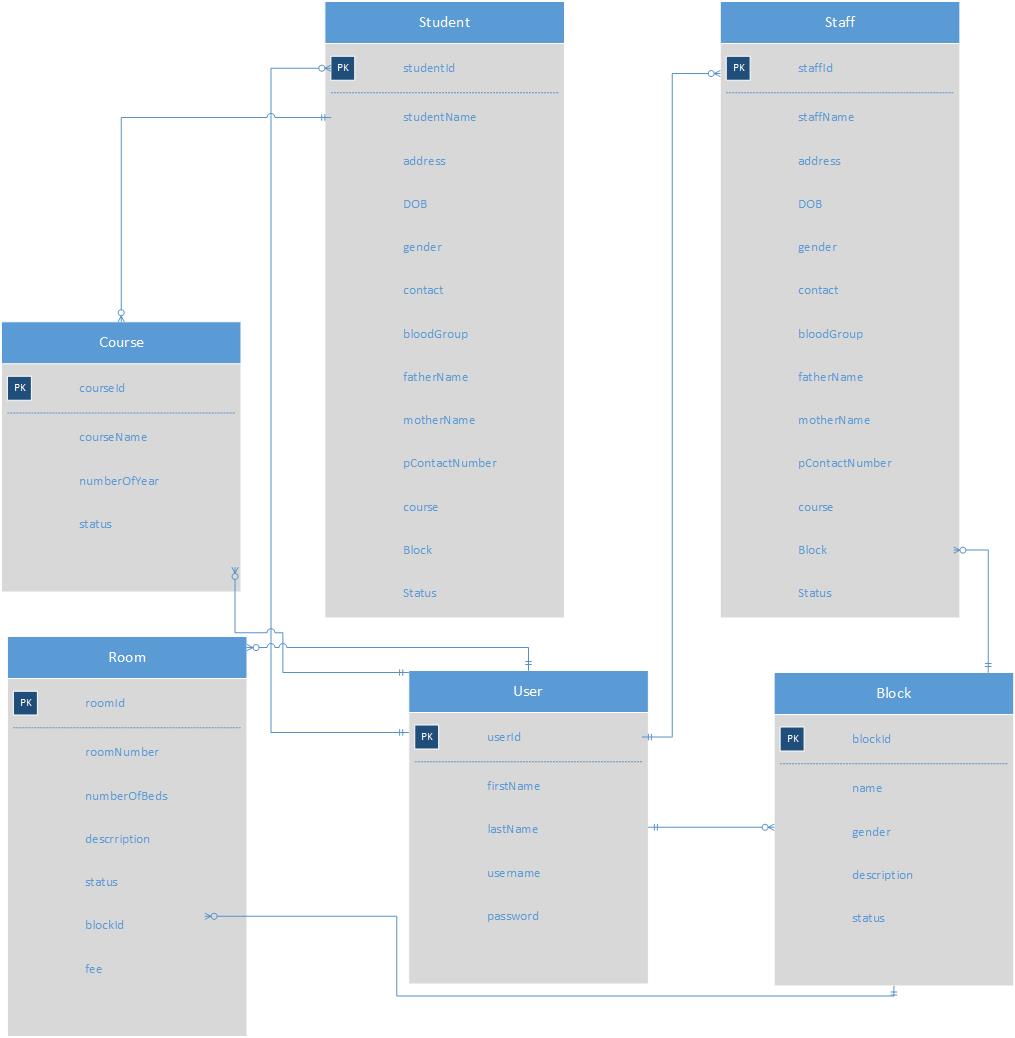


Fig: Entity Relationship diagram

## 3.4 Data Dictionary

Data dictionary refers to the table which contain data about data. The data dictionary of each entity are show below:-

**Student**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Datatype | PK/FK | Nullable |
| Student Id | INTEGER(10) | PK | NO |
| Name | STRING(150) |  | YES |
| Address | STRING(150) |  | YES |
| Date of Birth | STRING(30) |  | YES |
| Gender | STRING(15) |  | YES |
| Contact Number | STRING(15) |  | YES |
| Blood Group | STRING(5) |  | YES |
| Father Name | STRING(150) |  | YES |
| Mother Name | STRING(150) |  | YES |
| Parent’s Contact Number | STRING(15) |  | YES |
| Course | STRING(50) |  | YES |
| Block | STRING(30) |  | YES |
| Status | STRING(30) |  | YES |

**Staff**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Datatype | PK/FK | Nullable |
| Student Id | INTEGER(10) | PK | NO |
| Name | STRING(150) |  | YES |
| Address | STRING(150) |  | YES |
| Date of Birth | STRING(30) |  | YES |
| Gender | STRING(15) |  | YES |
| Contact Number | STRING(15) |  | YES |
| Blood Group | STRING(5) |  | YES |
| Father Name | STRING(150) |  | YES |
| Mother Name | STRING(150) |  | YES |
| Parent’s Contact Number | STRING(15) |  | YES |
| Designation | STRING(50) |  | YES |
| Block | STRING(30) |  | YES |
| Status | STRING(30) |  | YES |

**Room**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Datatype | PK/FK | Nullable |
| Room Id | INTEGER(10) | PK | NO |
| Room Number | STRING(10) |  | NO |
| Number of Beds | STRING(30) |  | YES |
| Description | STRING(150) |  | YES |
| Status | STRING(15) |  | NO |
| Block | STRING(15) |  | NO |
| Fee | INTEGER(10) |  | YES |

**Course**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Datatype | PK/FK | Nullable |
| Course Id | INTEGER(10) | PK | NO |
| Course Name | STRING(50) |  | NO |
| Number Of Year | INTEGER (10) |  | YES |
| Status | STRING(50) |  | NO |

**Block**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Datatype | PK/FK | Nullable |
| Block Id | INTEGER(10) | PK | NO |
| Block Name | STRING(50) |  | NO |
| Gender | STRING (15) |  | NO |
| Description | STRING(150) |  | YES |
| Status | STRING(15) |  | NO |

### 3.5. Behavioral Model

Behavioral models are models of the dynamic behavior of the system as it is executing. They show what happens or what is supposed to happen when a system responds to a stimulus from its environment.

## 3.5.1 Activity Diagram

Activity diagram is the graphical representation of activities of the system. It describes the dynamic aspects of the system. It is something like flowchart which show which activity is going to perform after one activity. Activity of the system is also known as operation of the system.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not.

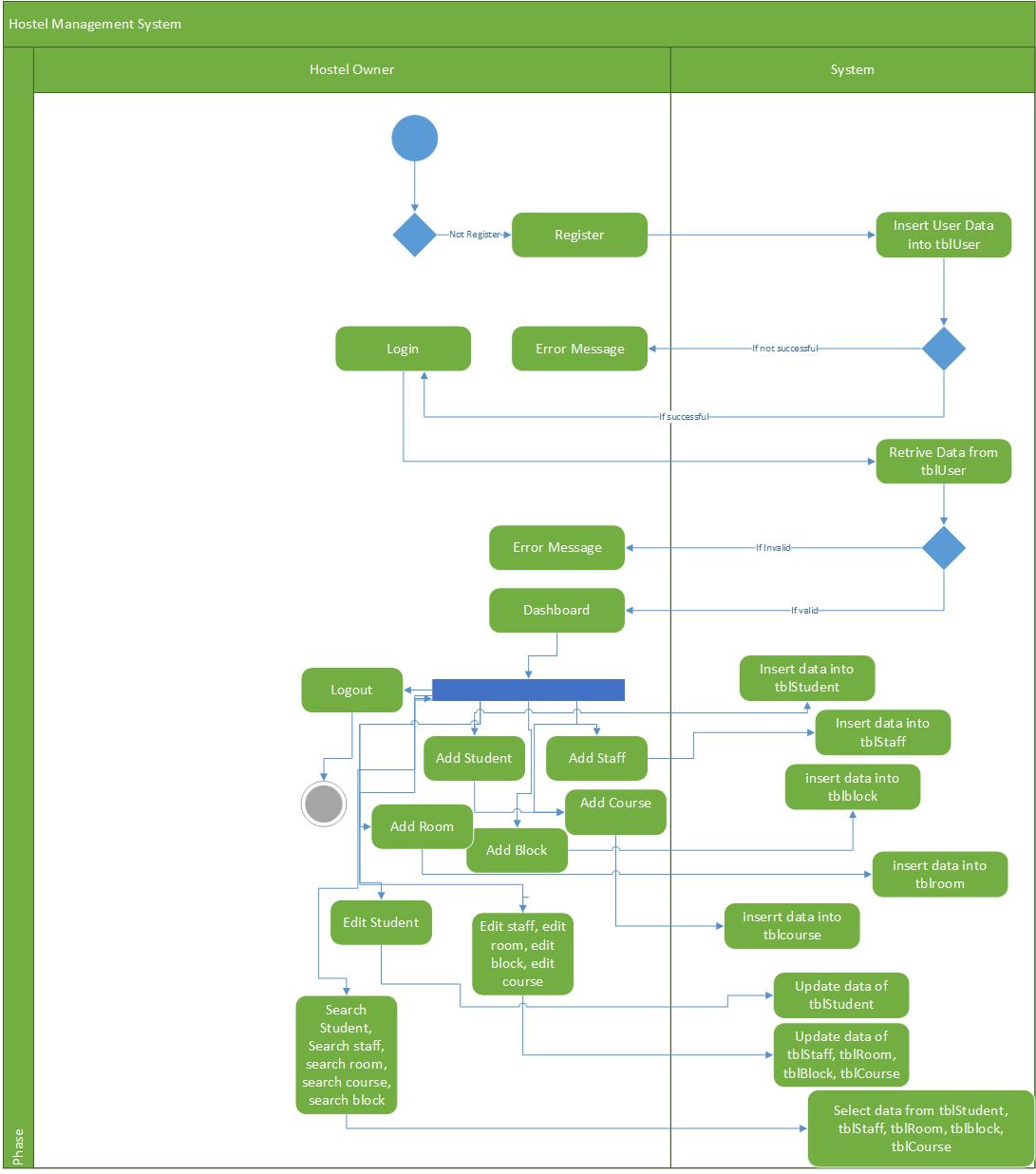


Fig: Activity Diagram

## 3.5.2 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

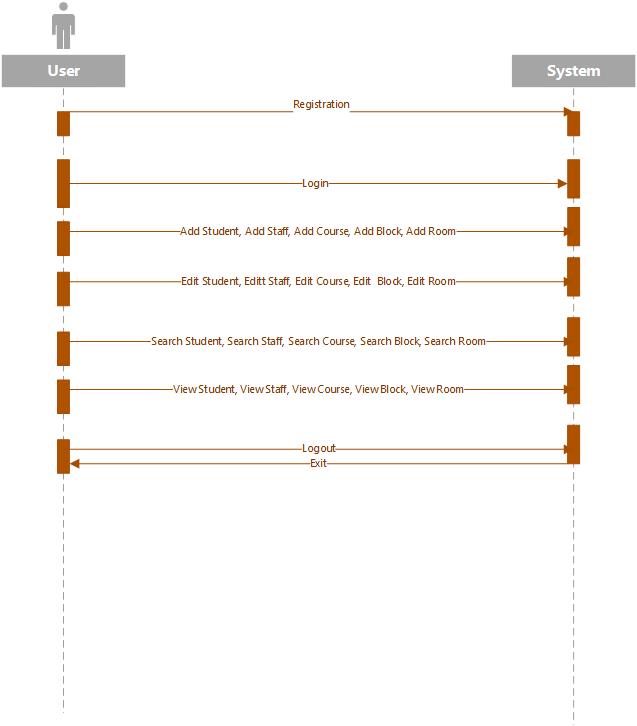


Fig: Sequence Diagram

## UI Design

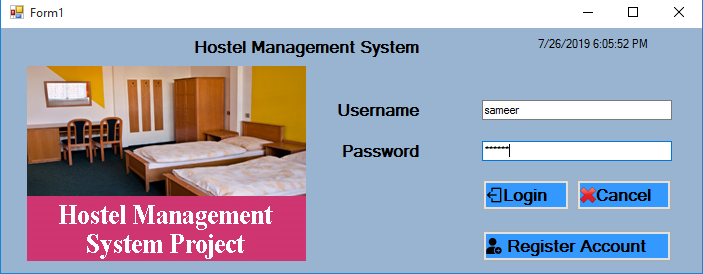


Fig: Login panel

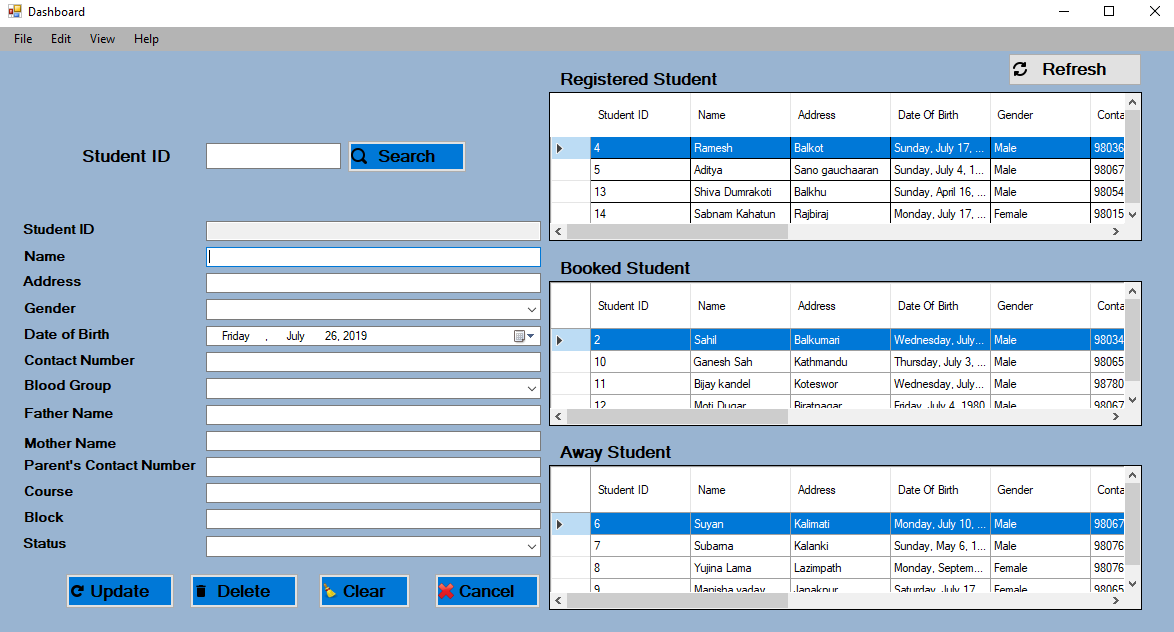


Fig: Dashboard

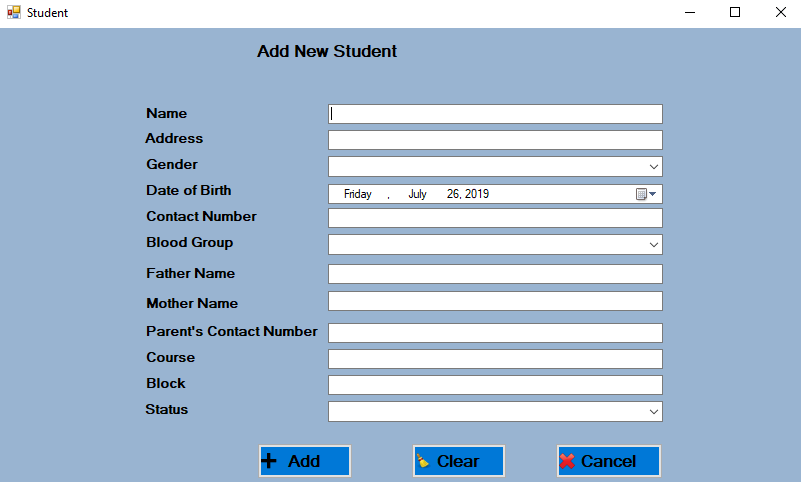


Fig: Add Student

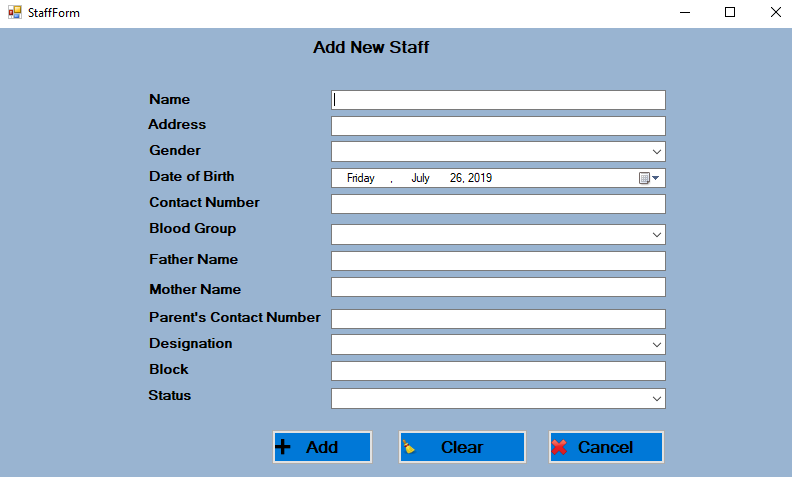


Fig: Add Staff

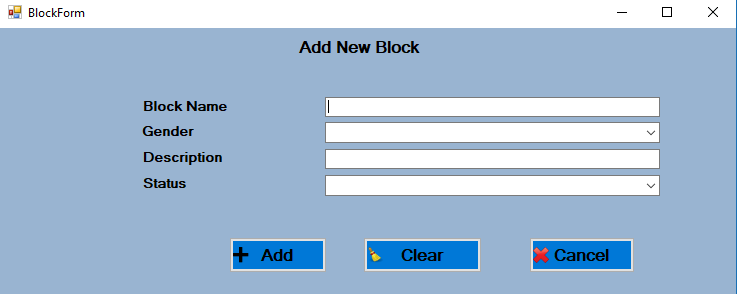


Fig: Add Block

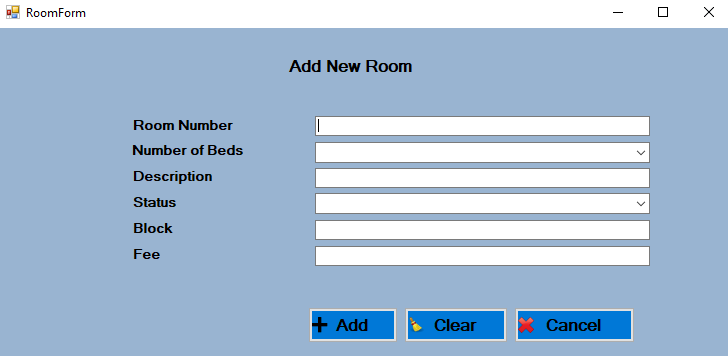


Fig: Add Room

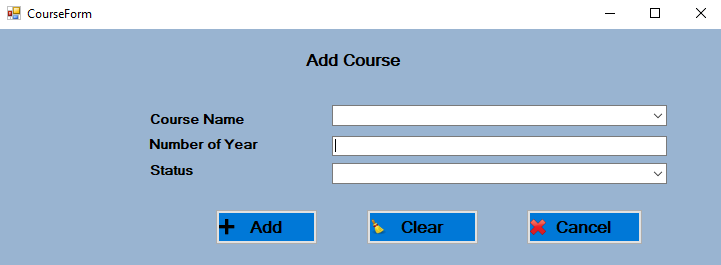


Fig: Add Course

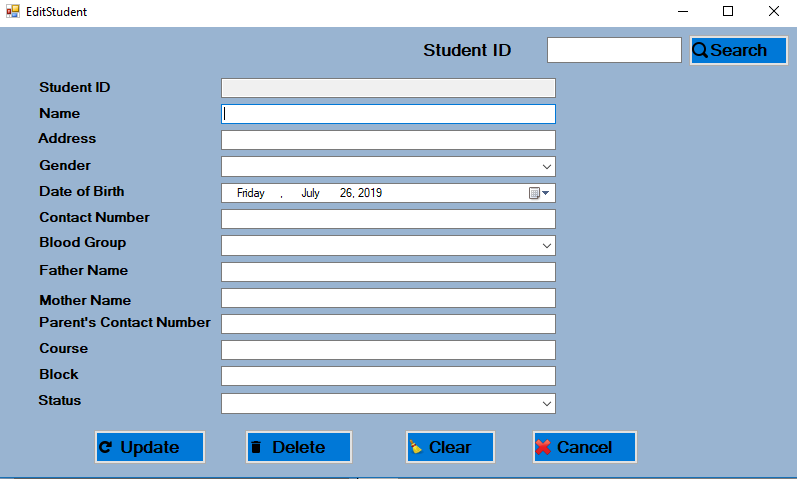


Fig: Edit Student

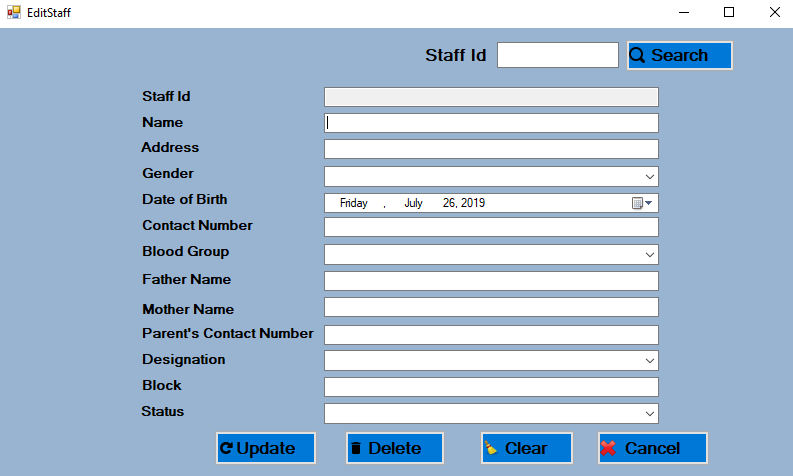


Fig: Edit Staff

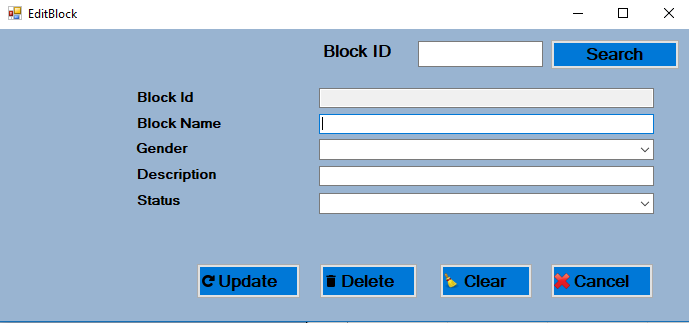


Fig: Edit Block

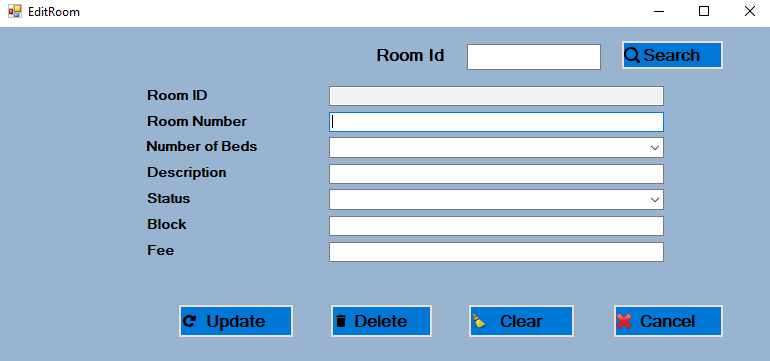


Fig: Edit Room

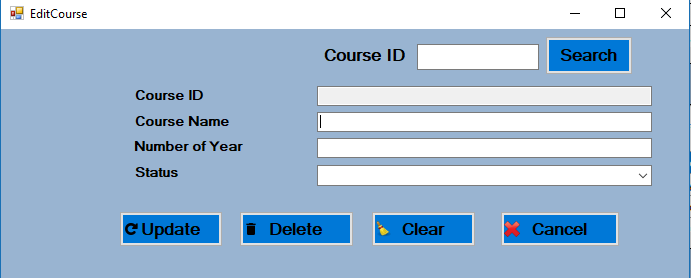


Fig: Edit Course

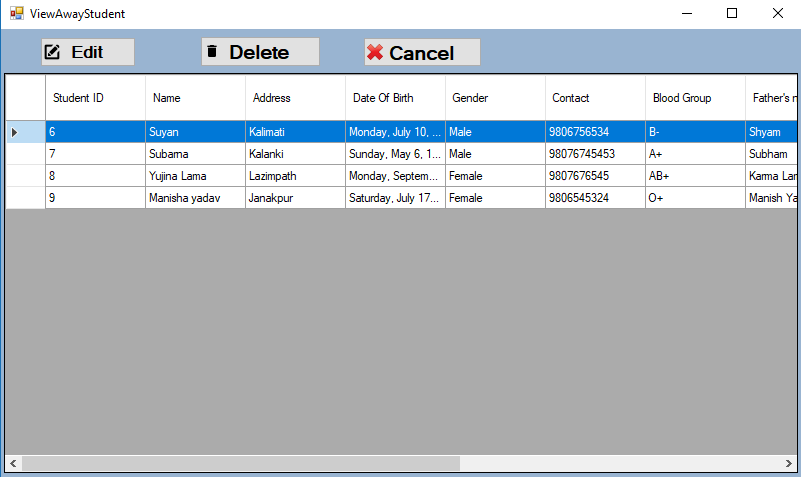


Fig: View Student

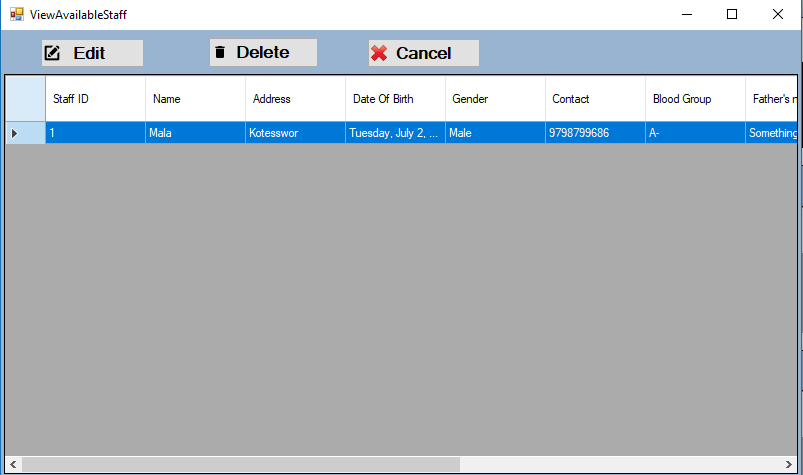


Fig: View Staff

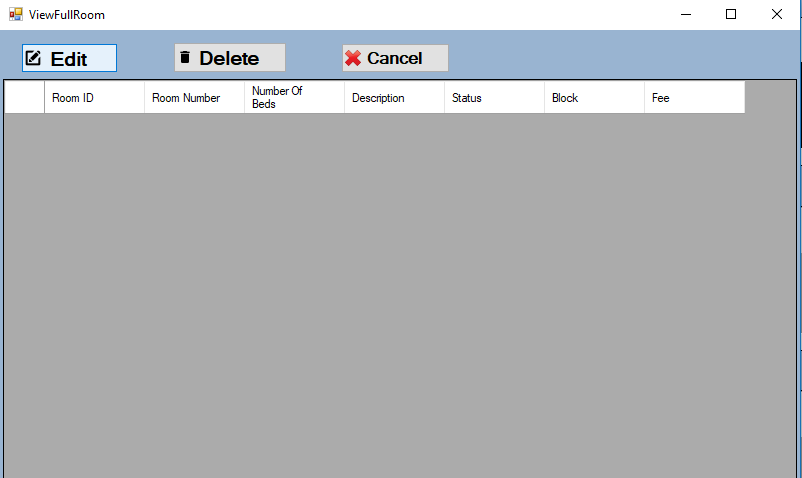


Fig: View Room

# Chapter 4: Implementation

## 4.1 Introduction

The process of application development comes under Implementation. In this process, developer convert the proposed system into real working system. Here I used dot net framework for the development of project. C sharp has been used as the programming language. The software I am developing is desktop application which means that it needs to install before running it. If there is need to work in different computer, then it needs to install on all the system separately.

## 4.2 Programming Language

C# has been used as the programming language for the development of application. Application has been developed under the dot NET Framework. C# is ranked as top most programming language in the recent year. It makes the development of application easy and efficient.

## 4.3 Development Environment

I have use visual studio for the development of project. Development environment is also known as IDE. IDE stands for integrated development environment. IDE helps in development of application easy and accurate.

For database MySQL has been used. To draw all the required class diagram Microsoft Visio has been used. Visio is one of the popular application to draw the diagram of software engineering.

## 4.4 Deployment Strategy

Deployment is the process of real installation of application in system. It defines the package of sub components which make up the application fully functional. There are different Procedure associated with application deployment. They are explained below briefly: -

* **Release:** It is process of competing all the operations to the computer System. It also involves in determination of resources required for the system to operate with tolerable performance and planning and/or documenting subsequent activities of the deployment process.
* **Installation and activation:** It is process of installing application into the system and creating shortcut, script for the execution of the software. There needs some product key for the activation purpose. Sometimes it involves configuration of the system – possibly asking end-user questions about its intended use.
* **Deactivation:** It is the process of deactivating the application when it is not according to the documentation or using it without entering product key.
* **Uninstallation:** It involves the process of removing application from the system.
* **Update:** When the application is added with some more additional features, application needs to update to accommodate the addition features into it.
* **Version tracking:** Version tracking helps the end user to find and install updates to the application.

## 4.5 User Training

It is the important part in the development of application. When the design of application completed for real field use. The user may unknown about its use as they are not technically fluent about the system. So, training should be provided to them to use the system. Different strategy can be used for user training like presenting the application in front of them, by interviews, workshop, etc.

# Chapter 5: Testing

## Unit Testing

Unit Testing refers to the testing the small units of application individually. Unit is small portion of the project. There are many units in a project which is to be tested separately. The main task of unit testing is to ensure that the product is designed as proposed previously. Unit testing accepts one or two inputs and generate only one output. The purpose of unit testing is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is method, which may belong to a base/super class, abstract class or derived/child class.

## Integration Testing

It is the type of testing which is performed after combining all the units of the application into single unit. The main purpose of this testing is to find out the fault or errors in the system when all the individual units are combined together. Test case are used for the process of unit testing. Test case table is shown below:-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Case** | **Expected Result** | **Actual Result** | **Pass / Fail** | **Remarks** | **Final Result** |
| 1 | Registration | Owner should be able to register with the system | Are able to register with system | Pass | No action taken | - |
| 2 | login | Owner should be able to login into the system | Owner can login into the system | Pass | No action taken | - |
| 3 | Add Student | Owner should be able to add new student | Are able to add student | Pass | No action taken | - |
| 4 | Add Staff | Owner should be able to add Staff | Owner can add new Staff | Pass | No action taken | - |
| 5 | Add Block | Owner should be able to add new block | Owner can add new block | Pass | No action taken | - |
| 6 | Add Room | Owner should be able to add new room | Owner Can add new room | Pass | No action taken | - |
| 7 | Add Course | Owner should be able to add new Course | Owner can add new Course | Pass | No action taken | - |
| 8 | Edit Student Information | Owner should be able to edit student information | Owner can edit student information | Pass | No action taken | - |
| 9 | Edit Staff Information | Owner Should be able to edit staff information | Owner can edit staff information | Pass | No action taken | - |
| 10 | Edit Block | Owner should be able to edit block | Owner can edit block | Pass | No action taken | - |
| 11 | Edit Room | Owner can edit room information | Owner can edit room information | Pass | No action taken | - |
| 12 | Edit Course | Owner should be able to edit course | Owner can edit course | Pass | No action taken | - |
| 13 | Delete Student | Owner should be able to delete student | Owner can delete student | Pass | No action taken | - |
| 14 | Delete Staff | Owner should be able to delete staff | Owner can delete staff | Pass | No action taken | - |
| 15 | Delete Block | Owner should be able to delete block | Owner can delete block | Pass | No action taken | - |
| 16 | Delete Room | Owner should be able to delete room | Owner can delete room | Pass | No action taken | - |
| 17 | Delete Course | Owner should be able to delete course | Owner can delete course | Pass | No action taken | - |
| 18 | Search Student | Owner should be able to Search student | Owner can search student | Pass | No action taken | - |
| 19 | Search Staff | Owner should be able to search staff | Owner can search staff | Pass | No action taken | - |
| 20 | Search Block | Owner should be able to search block | Owner can search block | Pass | No action taken | - |
| 21 | Search Room | Owner should be able to search room | Owner can search room | Pass | No action taken | - |
| 22 | Search  Course | Owner should be able to search course | Owner can search course | Pass | No action taken | - |
| 23 | View student according to status | Owner should be able to view student | Owner can view student | Pass | No action taken | - |
| 24 | View Staff according to status | Owner should be able to view staff | Owner can view staff | Pass | No action taken | - |
| 25 | View Room according to Status | Owner should be able to view room | Owner can view room | Pass | No action taken | - |
| 26 | View Block according to Status | Owner should be able to view block | Owner can view block | Pass | No action taken | - |
| 27 | View Course according to Status | Owner should be able to view course | Owner can view Course | Pass | No action taken | - |
| 28 | Logout | Owner should be able to logout from the system | Owner can logout from the system | Pass | No action taken | - |

**Table: Test cases**

# Chapter 6: Project Issues

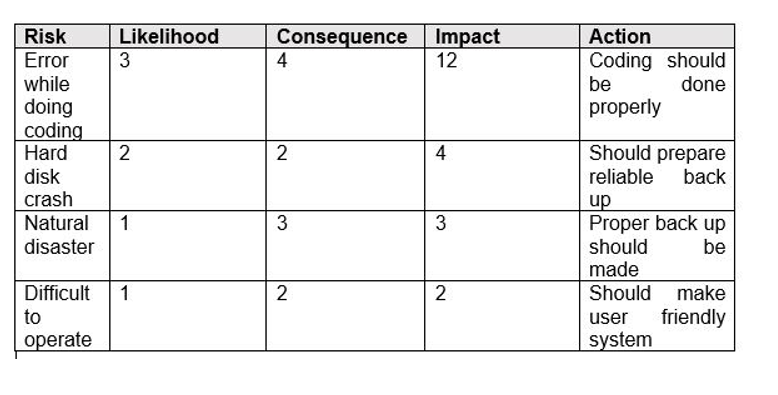
## Project Management

Project Management is represented as the spread of smearing precise methodology and standards to start, outline, implement and accomplish the method that new workouts or fluctuations are understood exclusive a connotation. Responsibility supervision is typical corporate management as consistent drive which is a determined organization, as it integrates creation new exertion packs to reach arranged conclusions or terminuses. Responsibility management is important subsequently it confirms what is being approved on, is correct, and will permit on reliable incentive in contradiction of the commercial occasion.

## Risk Management

It is the processes of examining, classifying and responding to any threats that appears done lifespan cycle of scheme to permit the scheme to stay on track and meet its impartial. A risk is whatsoever that might perchance scrape project’s period mark budget or act. Here we cast-off risks in command to recognize the danger and to preserve in an establishment method. In this technique we can abridged the risk which strength be revenue residence when emerging scheme.

Due to which we can resolve numerous extremely regarded intimidations that asset originate while emerging our scheme.



## Configuration management

Configuration management is a systems engineering process for establishing and maintaining consistency of a product’s performance, functional, and physical attributes with its requirements, design and operational information throughout its life. The CM process is widely used IT Service Company.

# Chapter 7

## Future Work

It is laid-back to extend the hostel management system that we have projected. An individual might perceive few of the dispensed, unissued or entirely the rooms agreeing to their drive. In upcoming days, we can appliance few specializations in HOSTEL MANAGEMENT SYSTEM. In this scheme it is likely to classify room charge for mid class pupils and underprivileged pupils. Few underprivileged pupils are assumed a specific franchise for the whole year.

# Conclusion

The Hostel Management System is a Tailor and accessible project for relevant Hostel. It has been intended to systematize, accomplish and stare next the whole dispensation of horizontal actual big hostel. It is accomplished of handling Query specifics, Student Specifics, Payment Specifics etc. The Hostel Management System is adapting and accessible software on behalf of Hostel which offer hostel info, hostel room info, and hostel interpretations data. Hostel Management System is proposing an extreme of constancy, cost-effectiveness and usable. It provides the maximum malleable and flexible values of hostel management system software resolutions for the relevant hostel.

# Appendix

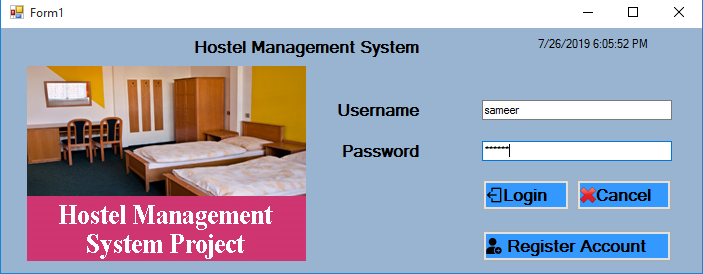


Fig: Login Form

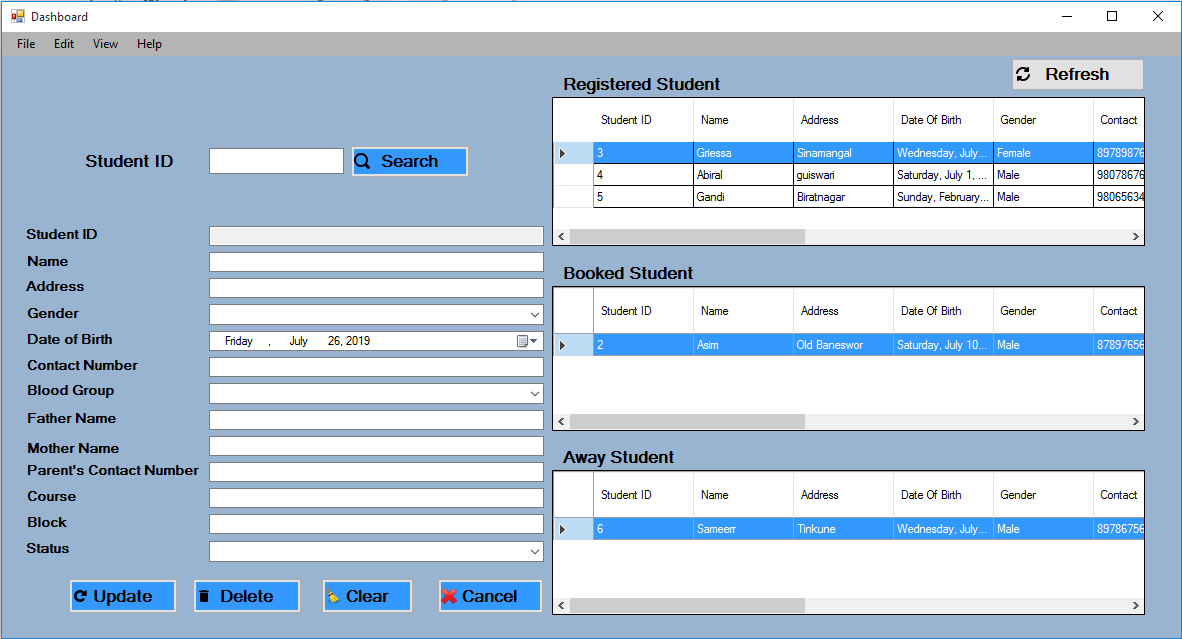


Fig: Dashboard

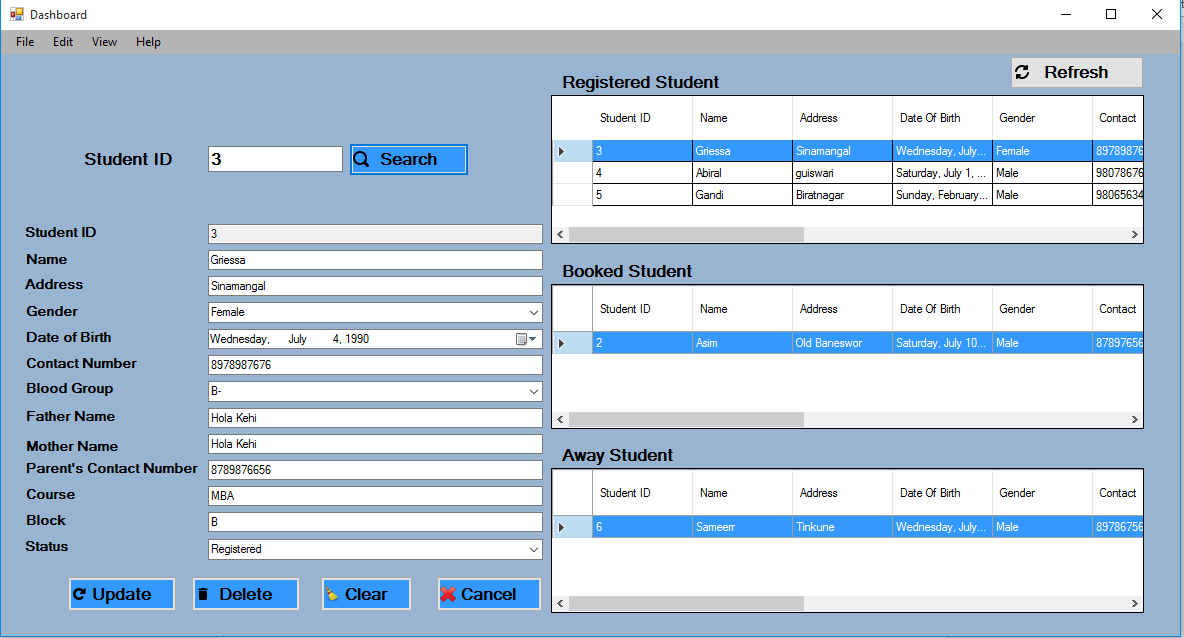


Fig: Searching Student whose ID is 3

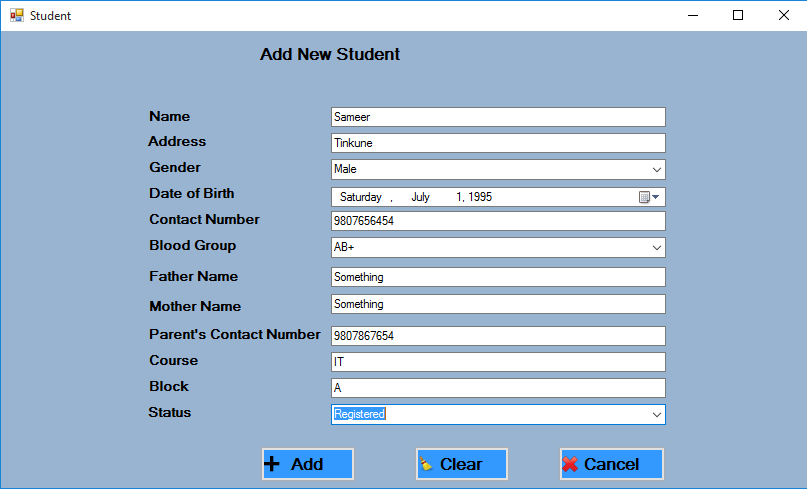


Fig: Adding new student

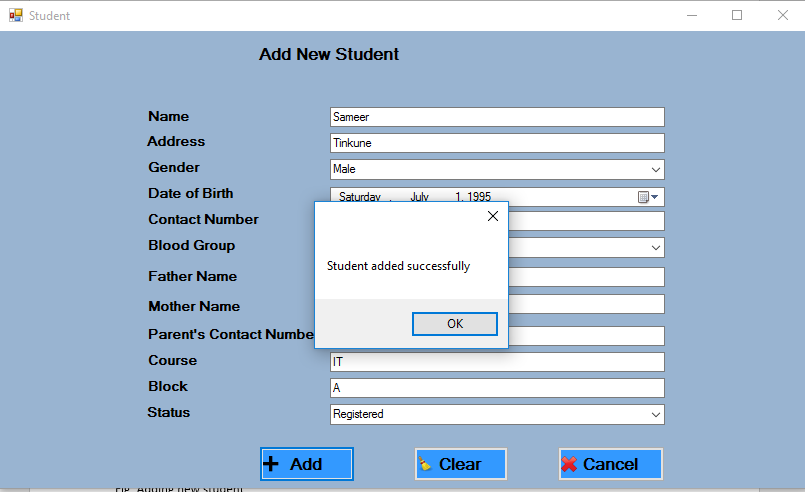


Fig: Successful Message after adding new student

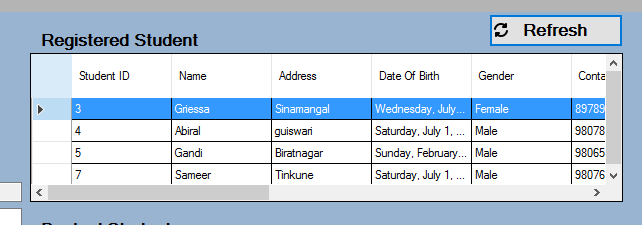


Fig: New added Student

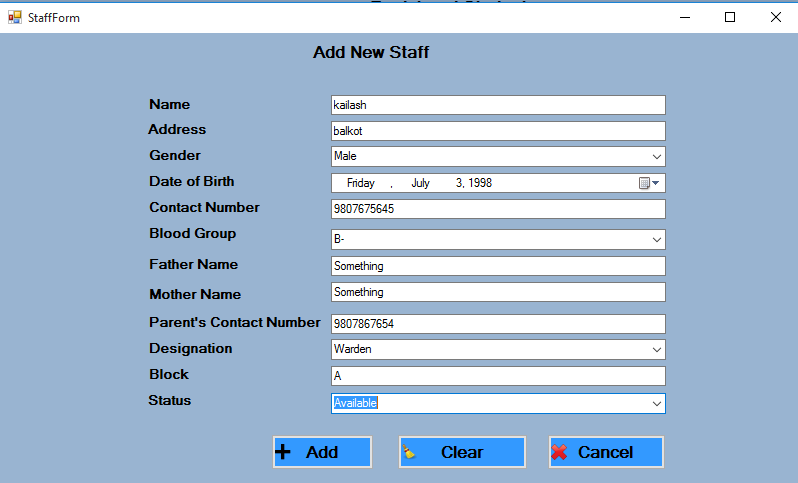


Fig: Adding new Staff

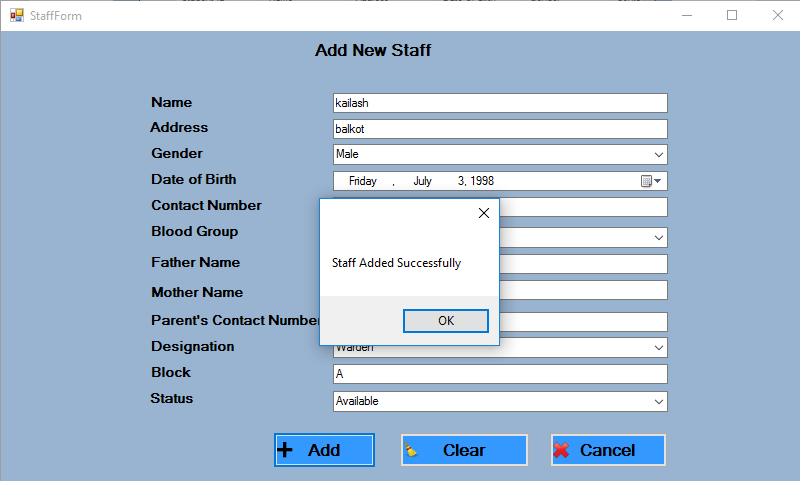


Fig: Successful message after adding new staff

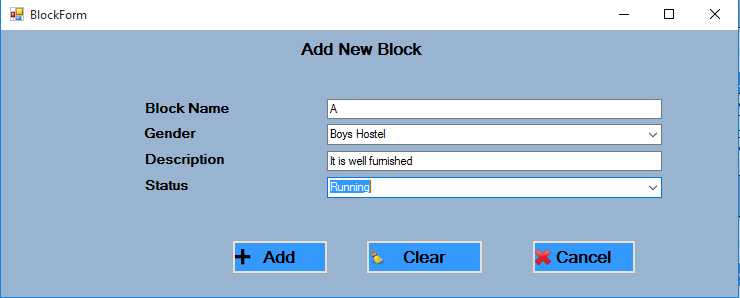


Fig: Adding new Block

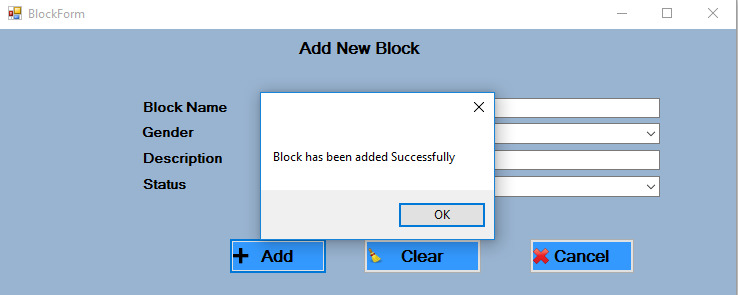


Fig: Successful message after adding Block

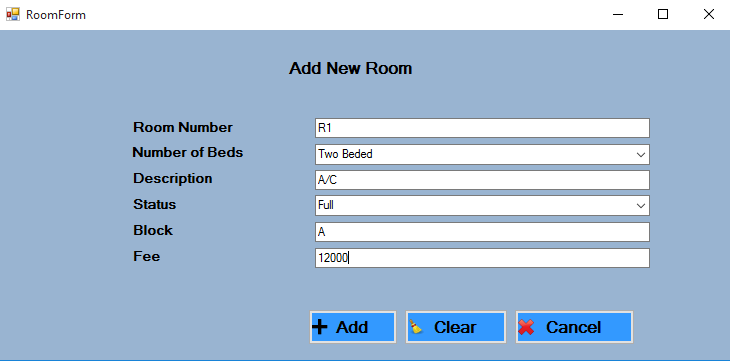


Fig: Adding new room in block A

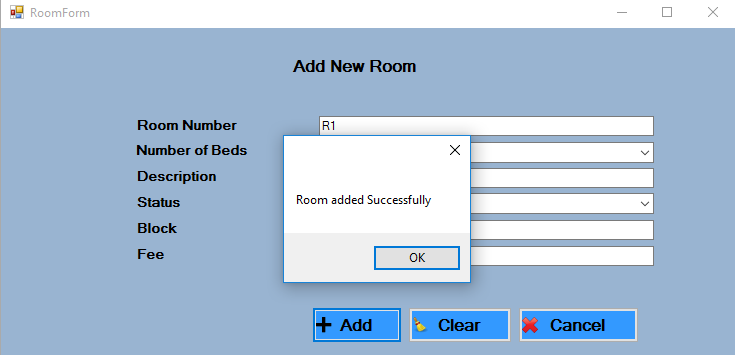


Fig: Successful message after adding new room

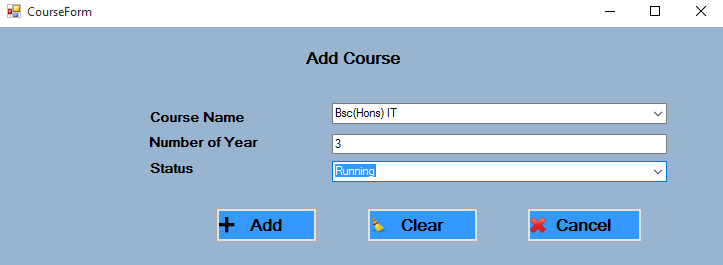


Fig: Adding new Course

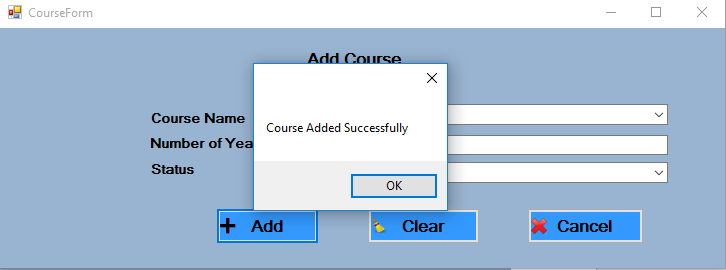


Fig: Successful message after adding new room

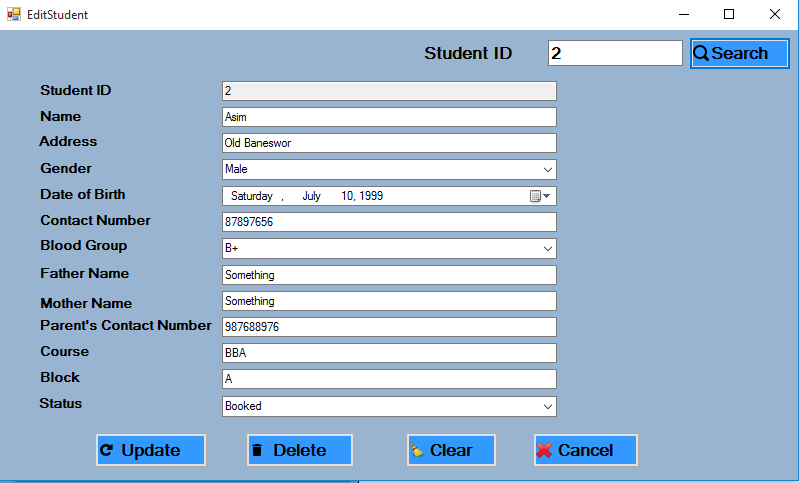


Fig: Searching student whose id is 2

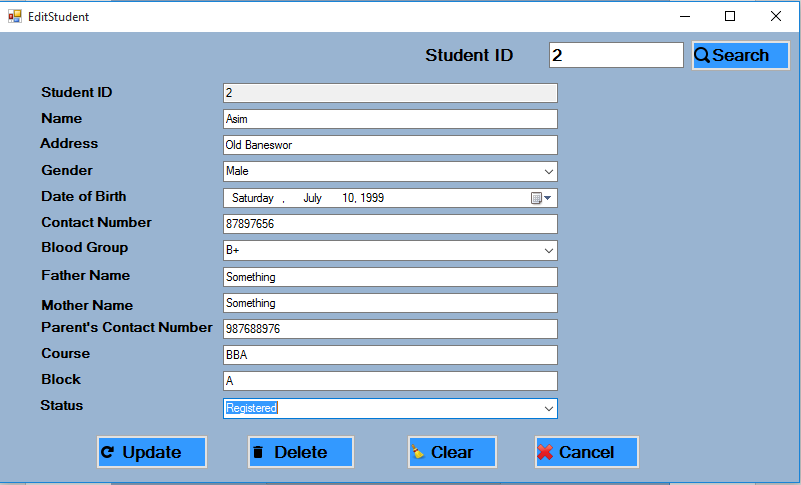


Fig: Updating Status of student whose id is 2

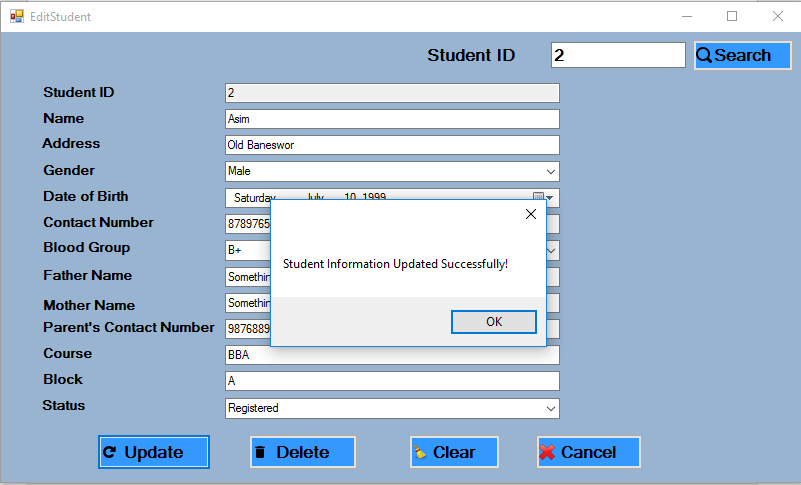


Fig: Successful message after updating student info

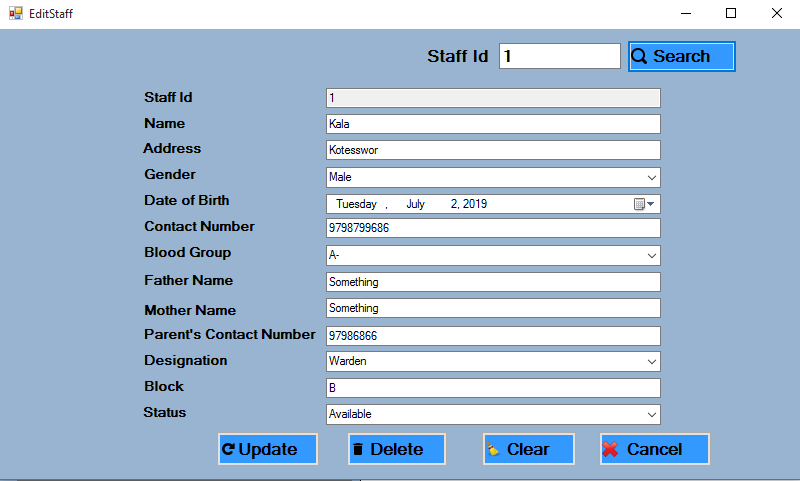


Fig: Searching staff whose id is 1

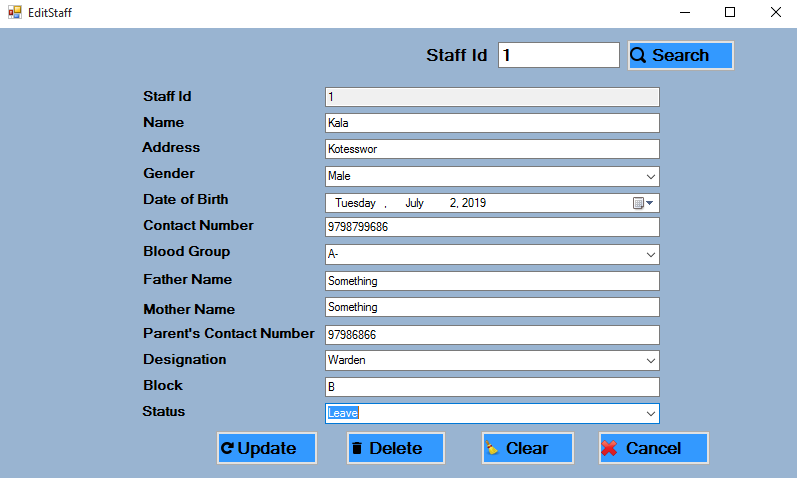


Fig: Updating staff info whose id is 1

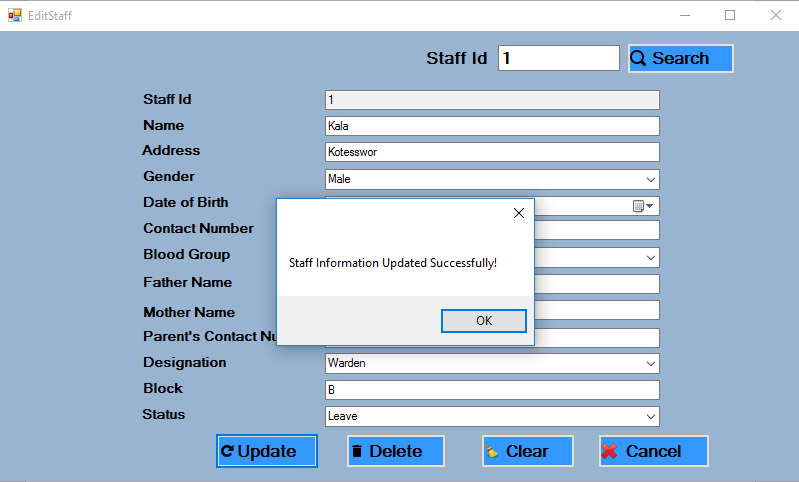


Fig: Successful message after updating information

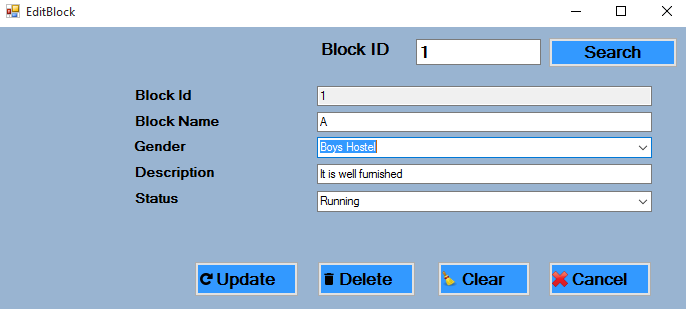


Fig: Searching block of id 1

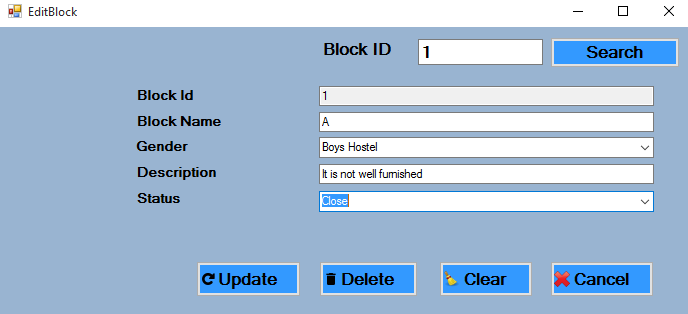


Fig: Updating block status

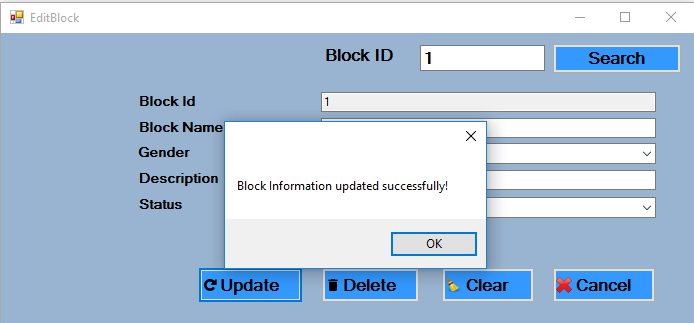


Fig: Successful message after updating block

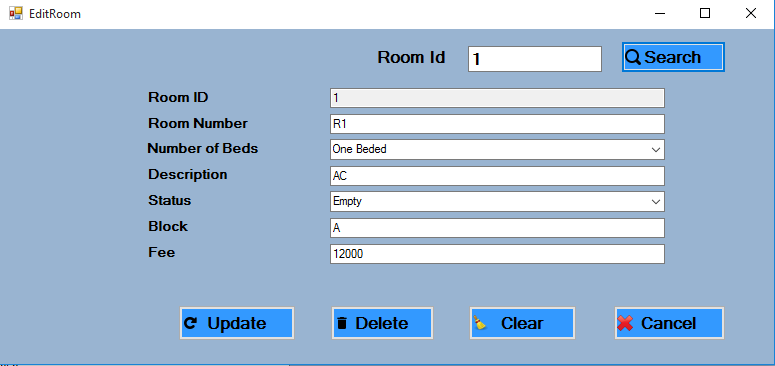


Fig: Searching room whose id is 1

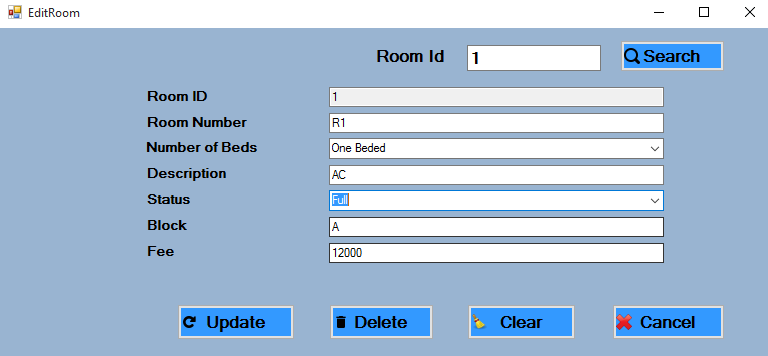


Fig: Updating it’s Status

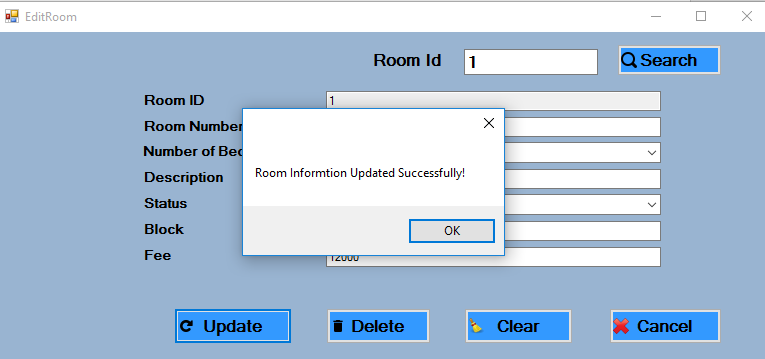


Fig: Successful message after updating room information

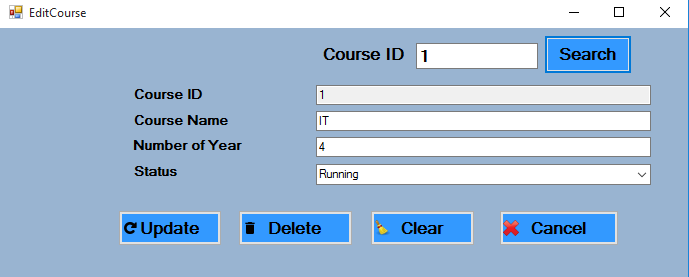


Fig: Searching course of id 1

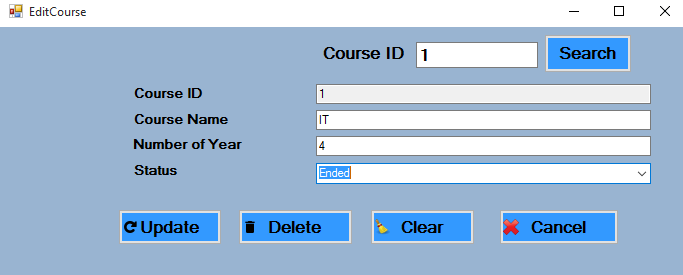


Fig: Updating Course Info

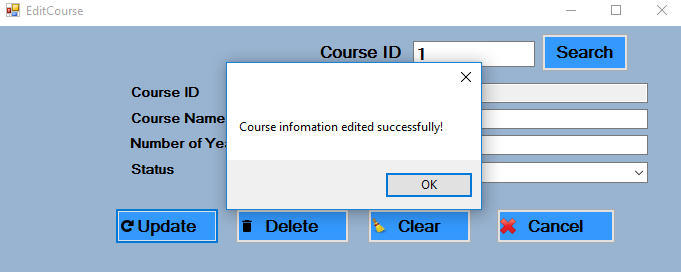


Fig: successful message of updating course

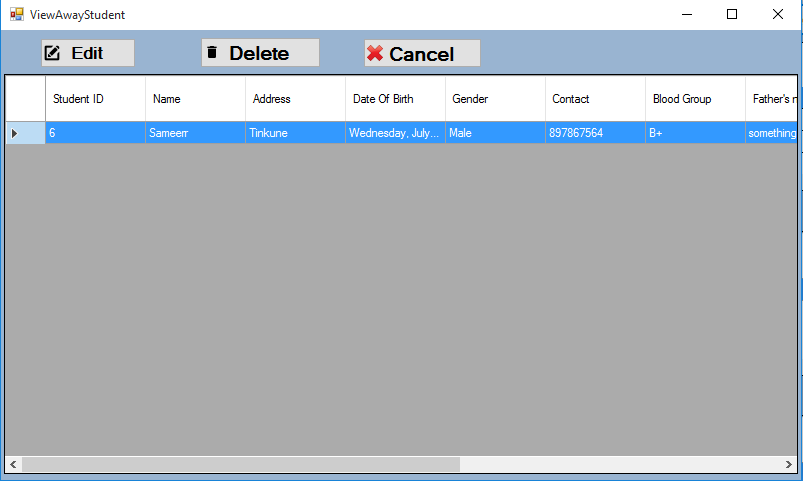


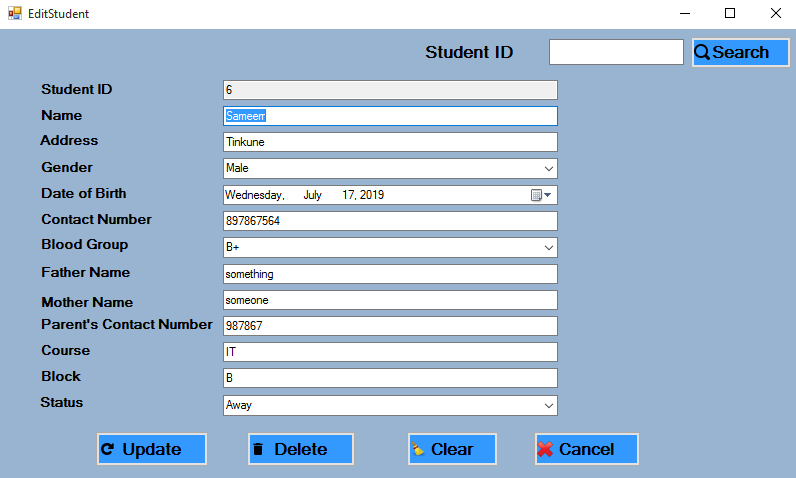
Fig: Viewing Away Student 

Fig: After clicking on Edit

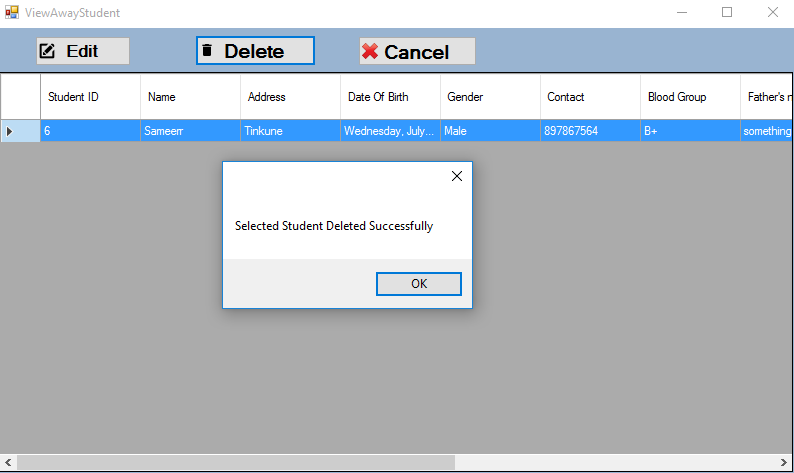


Fig: Deleting selected student from Grid View

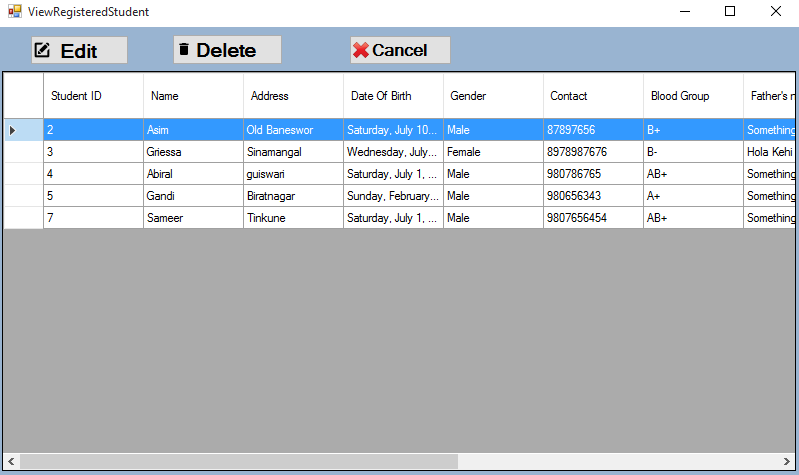


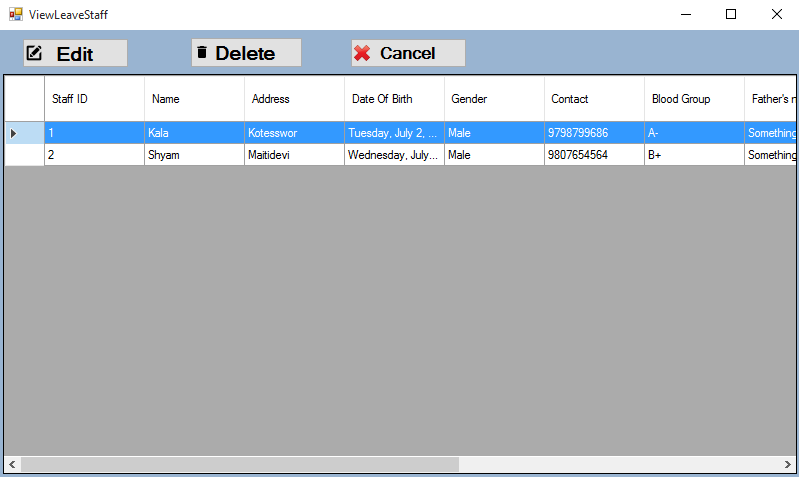
Fig: Viewing Registered Student 

Fig: Viewing Staff who are on leave

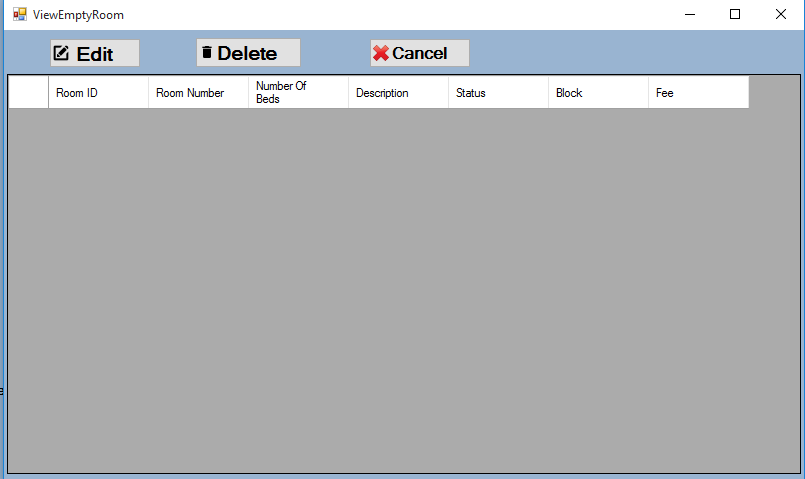


Fig: Viewing Empty Room

# References

* GeeksforGeeks. (2019). *Unified Modeling Language (UML) | Sequence Diagrams - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/ [Accessed 15 Jul. 2019].
* Sebokwiki.org. (2019). *System Architecture - SEBoK*. [online] Available at: https://www.sebokwiki.org/wiki/System\_Architecture [Accessed 14 Jul. 2019].
* WhatIs.com. (2019). *What is use case diagram (UML use case diagram)? - Definition from WhatIs.com*. [online] Available at: https://whatis.techtarget.com/definition/use-case-diagram [Accessed 16 Jul. 2019].
* Ifnoss Campus Management System for University, College, School and Institute. (2019). *University College Hostel Management System | IFNOSS*. [online] Available at: http://www.ifnoss.com/edu/college-university-software/hostel-management-system.aspx [Accessed 17 Jul. 2019].
* SearchMicroservices. (2019). *What is a class diagram? Definition from WhatIs.com.*. [online] Available at: https://searchmicroservices.techtarget.com/definition/class-diagram [Accessed 18 Jul. 2019].
* www.tutorialspoint.com. (2019). *UML - Activity Diagrams*. [online] Available at: https://www.tutorialspoint.com/uml/uml\_activity\_diagram [Accessed 20 Jul. 2019].
* Software Testing Material. (2019). *Software Testing - Definition, Types, Methods, Approaches*. [online] Available at: https://www.softwaretestingmaterial.com/software-testing/ [Accessed 23 Jul. 2019].