**Software Requirements Specification**

**for**

**<Centralized Complaint Module>**

**<Jan 26, 2023>**

**Team Members:**

**Web-Developer-1: MAHENDRA DODWARIYA (20BCS123)**

**Web-Developer-2: UJJAWAL LELE (20BCS234)**

**Database Manager: RITESH BHANDARIA (20BCS183)**

**Documentation: SASANK B.V.M(20BCS060)**

**UI/UX Designer: MIRIYALA TARUN KUMAR (20BCS132)**

**Table of Contents**

**Table of Contents………………………………………………………………………………………………………….2**

**Revision History… ………………………………………………………………………………………….3 1.Introduction…………………………………………………………………………………………………4**

1.1 Introduction……………………………………………………………………………………4

1.2 Intended Audience and Reading Suggestions………...……………………………………….4

1.3 Product Scope………..……………………………………………………………………..…4

1.4 References …………...………………………………………………………………………..4

**2.Overall Description………………………………………………………………………………………………….5**

2.1 Product Functions….…………………………………………………………………………5

2.2 User Classes and Characteristics………………………………………………….………….5

2.3 Software Requirements Specification………………………………………………….…….5

2.4 Design and Implementation Constraints …………………………………………………….6

2.5 SRS offering..………………………………………………………………………….…….6

**3. External Internal Requirements………….……………………………………………………………...7**

3.1 System Interfaces ……….……………………………………………………………….….7

3.2 User Interfaces ………….……………………………………………………………….….7

3.3 Software Interfaces………...………………………………………………………………..7

3.4 Communication Interfaces...……………….………………………………………………..7

3.5 Operations…………………………..……………………………………………………….7

**4. System Features (Functional Requirements).........................................................8**

4.1 Use Case Diagram…………………………………………………………………………………………8

4.2 FR 1………………………………………………………………………..…………………9

4.3 FR2………...………………………………………………………………………….……..9

4.4 FR 3……………………………………………………………………………………….…9

4.5 FR 4……………………………………………………………………………………….…10

4.6 FR 5………………………………………………………………………………………….10

4.7 FR 6…………………...……………………………………….…………………………………..10

4.8 FR 7……...…………………………………………………….…………………………………..11

4.9 FR 8…………………...……………………………………….…………………………………..11

4.10 FR 9…………………...……………………………………….…………………………………12

4.11 FR 10…………………...……………………………………….…………………………………12

4.12 FR 11…………………………...……………………………….…………………………………12

4.13 FR 12……………………………..…………………………….…………………………………13

4.14 FR 13……………………………..…………………………….…………………………………13

4.15 FR 14…………………………..……………………………….…………………………………14

4.16 FR 15…………………………..……………………………….…………………………………14

4.17 FR 16……………………………….………………………….…………………………………14

4.18 FR 17…………………….…………………………………….…………………………………15

**5. Non-Functional Requirements………………………………………………………………………………16**

5.1 Ensuring security……………………..…………………………….………………..16

5.2 Performance………………………………………………….………………………16

5.3 Reliability………………………..……………………….…………………………..16

5.4 Usability …….………………………………………….……………………………16

5.5 Integrity……….……………………………………….…………………………….16

**Revision History**

| **Name** | **Date** | **Reason of Changes** | **Version** |
| --- | --- | --- | --- |
| Updation in usecases | 26-01-2023 | Some usecases needs to be updated. |  |
| Created new usecases and diagram for the actor Section Incharge | 14-02-2023 | Incorporated new actor Section Incharge |  |

**1. Introduction**

**1.1 Introduction**

The Centralized complaint system combines the issues related to Computer Center, Hostels and Cleanliness under one roof. This software aims at solving problems of students as well as staff as soon as possible by providing a platform to connect them directly to the Warden/Caretaker/Supervisor of the respective departments.

It will allow immediate addressal of complaints.

**1.2 Intended Audience and Reading Suggestions**

This software requirement specification is intended for all project members associated with this project .Users(i.e. Students/Staff/Faculty), Hostel CareTaker ,CC Admin, Cleanliness Supervisor, Section Incharge and other parties that have interest in this project can also use this document to gain a better understanding of the software .This Specification is organized into several sections that can be read and referenced as needed.

**1.3 Product Scope**

The software will allow registration of complaints by student. These complaints can be viewed by respective authority and workers can also be assigned accordingly to resolve the complaints.

**1.4 References**

Book : An Integrated Approach to Software Engineering by Pankaj Jalote.

IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended

IEEE Std 830-1984 — IEEE Guide to Software Requirements Specifications

Website:<http://cc.iiitdmj.ac.in/>

**2.** **Overall Description**

Complaining in the respective offices is time consuming. Furthermore, there are several disadvantages to it as the complaints are lost without being serviced, so this software is being designed so that all the complaints are at one place and the complainant can track them as well as communicate with the service providers.

Combining all these functionalities into software reduces service time and increases efficiency. This software is a replacement of an existing system but is customized for the needs of **Computer Center, Hostels and Cleanliness** of IIITDMJ.

**2.1 Product Functions**

* This system will completely automate the process of registering complaints.
* The complaints will reach to the authority in charge immediately.
* Solve issues very easily and quickly.
* If the people are unaware about the problem, they can upload snapshot or picture of the issue.
* Related persons can also easily know the problems related to their department just by checking their notification panel.
* Students can find the progress of the complaint.

**2.2 User Classes and Characteristics**

Cc -administrator sees the cc issues

Warden-Hostel issues.

Supervisor-Cleanliness issues

Section Incharge - Assign Workers to complaints

Faculty, student-complaints regarding computer Centre, Hostel and cleanliness.

**2.3 Operating Environment**

Any web browser irrespective of the hardware and operating system.

**2.4 Design and Implementation Constraints**

* Django framework
* Python
* MYSQL
* HTML/CSS
* JavaScript

**2.5 Software Requirements Specification**

SRS is the starting point of the software development activity. It is the means of translating the ideas of the minds of the clients into a formal document.

The role of SRS is as follows:

* *SRS is the basis of agreement between the user and supplier.*
* *It helps user to understand their needs when the developer provides them with perspective.*
* *It provides reference of validation*
* *A high-quality SRS ensures high quality software.*
* *A good SRS reduces development time*

**3. External Interface Requirements**

**3.1 System Interfaces**

The client systems should be able to share the data available in the database through the network connection. The System uses the database stored on that system or other system to retrieve the complaints and to store a new complaint.

**3.2 User Interfaces**

The GUI should be designed in such a way that it is easy to use for the user. It should be user friendly and very interactive. The scrolling should be as minimum as possible and everything should be available in an organized and formal way. There should be mechanisms for displaying error messages if unauthorized users try to access it, etc. There should be separate categories like Complaints for Carpenter, Complaints for Plumber, Complaints for Electrician so that user can easily make complaint in the required category. The menus should be organized according to priority with more important ones coming first so that user can easily locate them on interface.

**3.3 Software Interfaces**

Operating System: Windows

Client Software: Any Web Browser

Communication Network: Internet

**3.4 Communication Interface:**

The SRS can be accessed from college’s intra web server. It will be linked with college’s web server since it is to be used in college only. In future, it could be accessed from global web server so that students can access the software from their homes itself.

**3.5 Operations:**

The student logins to the software using his/her username and password.

The password will be given to user when he/she registers himself at the software or added by the admin.

If a student wishes to make a complaint, then he/she should follow following steps:

Login to the Centralized Complaint Management Software with their user id Roll Number in case of students and guest in case of guests and password.

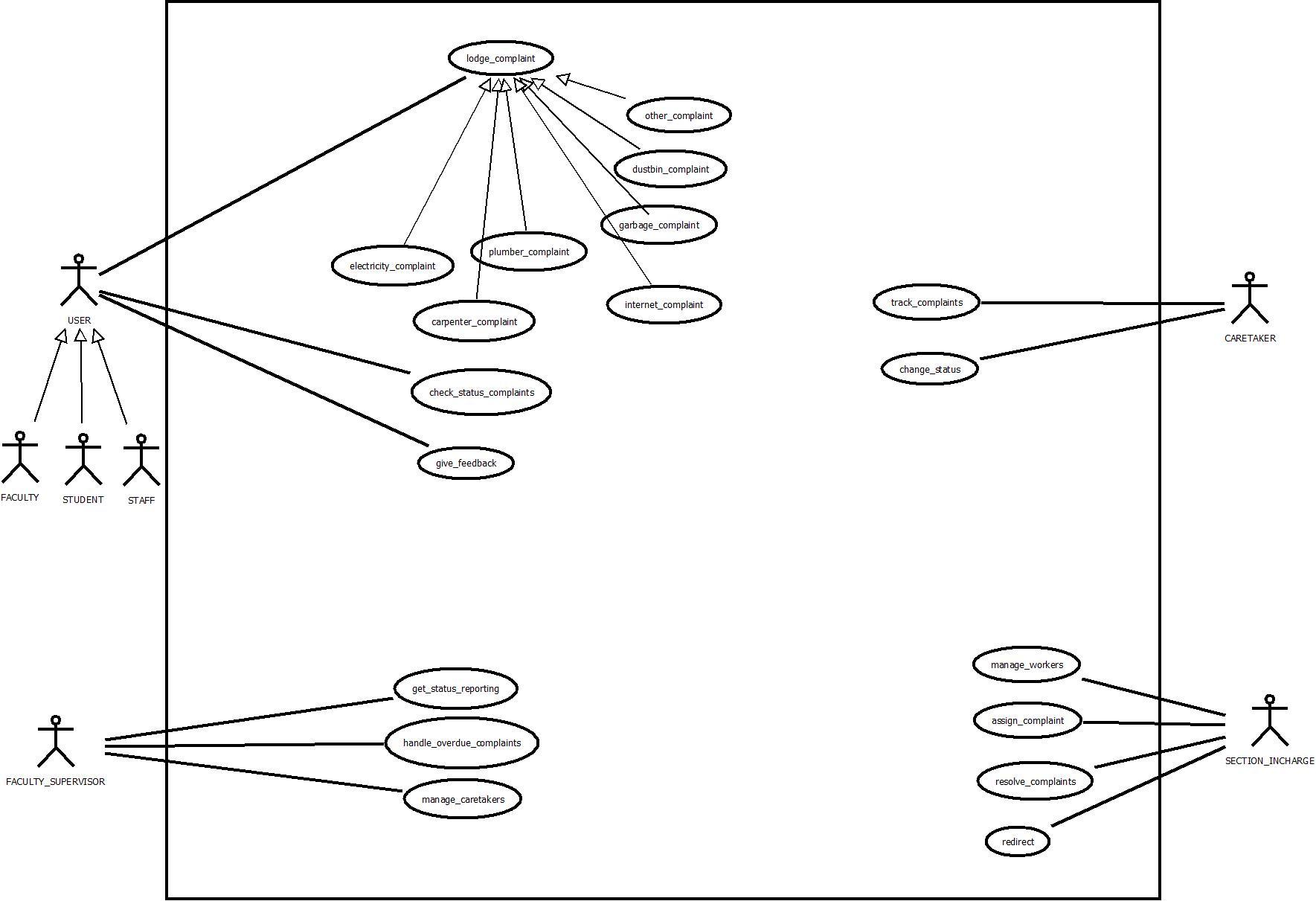
Make a complaint by selecting a category (which refers to the respective department to which the complaint is to be made) and then filling in a form and providing details like what is the complaint, student’s room number, and suggestions.

The software will already have details of the student like Name, Contact Number, Address, etc (these were entered by student during registration), so the software will directly fetch these details from database.

After registering the complaint, the student will be given a unique complaint Id which the student can use to track status of his/her complaint.

**4.1 System Features**

We specify the functional requirements for the module using use cases. The figure represents the use case diagram for the module features.



**4.2 Use case #1**

| **UC ID** | UC#1 | |
| --- | --- | --- |
| **Use Case Name** | **save\_comp** | |
| **Actor** | Student, Staff | |
| **Pre-Condition** | The user must fill the all the necessary fields in the complaint page. | |
| **Main Flow** | 1. | User fills all the necessary columns and clicks on the submit button. |
| 2. | System displays a message that complaint has been successfully received. |
| **Post-Condition** | The complaint is successfully received by the system and stored in the database. | |

**4.3 Use case #2**

| **UC ID** | UC#2 | |
| --- | --- | --- |
| **Use Case Name** | **submitfeedback** | |
| **Actor** | Student, Staff | |
| **Pre-Condition** | The user must file a complaint. | |
| **Main Flow** | 1. | User clicks on the feedback option at the top. |
| 2. | System displays a range of rating from one to five and a feedback text box. |
|  | 3. | User gives the necessary rating and feedback in the feedback textbox and clicks on submit button. |
| **Post-Condition** | The feedback gets added to the database and caretaker gets notified. | |

**4.4 Use case #3**

| **UC ID** | UC#3 | |
| --- | --- | --- |
| **Use Case Name** | **check\_status\_complaints** | |
| **Description** | After logged-in system displays the same page where he/she complaints and another options are available showing Complaint History. | |
| **Actor** | Student, Staff | |
| **Pre-Condition** | User must be logged-in. | |
| **Main Flow** | 1. | User choose the Complaint History to check the status of the complaints. |
| 2. | System displays the page showing the status of all the complaints i.e whether the complaint has been addressed by the caretaker or not with the time and date when the user had given the complaint. |
|  | 3. | User can see which of his complaints are being resolved, kept on-hold, complaints that are resolved and complaints that are declined. |
| **Alternate Flow** | NILL | |

**4.5 Use case #4**

| **UC ID** | UC#4 | |
| --- | --- | --- |
| **Use Case Name** | **give\_feedback** | |
| **Description** | User has the option to give feedbacks to the service after his/her complaint has been resolved. | |
| **Pre-Condition** | User must be logged-in. | |
| **Main Flow** | 1. | User choose the feedback option given on the first page. |
| 2. | System shows the next page. |
| 3. | User choose the type of feedback i.e whether the feedback is about  CC(Library+CC), hostel, residential or garbage. |
| 4. | User gives the feedback. |
| 5. | User submits the feedback. |
| 6. | System displays the initial page. |
| **Post-Condition** | Feedback is successfully received by the system and stored in the database. | |
| **Alternate Flow** | NILL | |

**4.6 Use case #5**

| **UC ID** | UC#5 | |
| --- | --- | --- |
| **Use case Name** | **supervisor** | |
| **Description** | This is the page for the supervisor, where he/she can see the list of caretakers available currently along with the list of the complaints that are resolved, Unresolved and Overdue. | |
| **Actor** | Faculty\_supervisor | |
| **Pre-Condition** | The Faculty\_superviser must be logged-in. | |
| **Main Flow** | 1. | System displays the list of available caretakers. |
| 2. | System displays options Resolved Complaints in the bottom left corner. |
| 3. | System displays option Unresolved Complaints in the bottom left corner pending and onhold. |
| 4. | System displays option Overdue Complaints in the bottom left corner. |
| **Post-Condition** | Supervisor may click on his/her desired button. | |
| **Alternate Flow** | NILL | |

**4.7 Use case #6**

| **UC ID** | UC#6 | |
| --- | --- | --- |
| **Use case Name** | **caretaker\_id\_know\_more** | |
| **Description** | Superviser can see the list of complaints under a caretaker. | |
| **Actor** | Faculty\_supervisor | |
| **Pre-Condition** | The Faculty\_superviser must be logged-in and must click on any particular caretaker name in the View Caretaker button. | |
| **Main Flow** | 1. | Supervisor clicks on the desired caretaker. |
| 2. | System displays a new page of list of complaints under the caretaker. |
| 3. | System shows the complainer details and the status of the complaint. |
| **Post-Condition** | NILL | |
| **Alternate Flow** | NILL | |

**4.8 Use case #7**

| **UC ID** | UC#7 | |
| --- | --- | --- |
| **Use case Name** | **feedback\_super** | |
| **Description** | Supervisor see’s the feedback given by the complainer. | |
| **Actor** | Faculty\_supervisor | |
| **Pre-Condition** | The Faculty\_superviser must be logged-in. | |
| **Main Flow** | 1. | Supervisor clicks on the option Resolved Complaints. |
| 2. | System displays a new page. |
| 3. | Supervisor clicks on the green Feedback button. |
| 4. | System shows the Feedback given to a particular complaint along with the details of the Complainer. |
| **Post-Condition** | NILL | |
| **Alternate Flow** | NILL | |

**4.9 Use case #8**

| **UC ID** | UC#8 | |
| --- | --- | --- |
| **Use case Name** | **detail3** | |
| **Description** | Supervisor can see the detailed version of the compliant. | |
| **Actor** | Faculty\_supervisor | |
| **Pre-Condition** | The Faculty\_superviser must be logged-in. | |
| **Main Flow** | 1. | Supervisor clicks on the option unresolved complaints. |
| 2. | System displays a new page. |
| 3. | System shows the list of complaints. |
| 4. | Supervisor clicks on the green i button |
| 5. | System shows the detailed view of the complaint. |
| **Post-Condition** | NILL | |
| **Alternate Flow** | NILL | |

**4.10 Use case #9**

| **UC ID** | UC#9 | |
| --- | --- | --- |
| **Use case Name** | **add\_workers** | |
| **Description** | Section Incharge can manage the workers | |
| **Actor** | section\_incharge | |
| **Pre-Condition** | The Section Incharge must be logged-in.  In case to delete any worker, he/she must not be involved in any unresolved complaint. | |
| **Main Flow** | 1. | Section Incharge clicks on the option manage workers. |
| 2. | System displays a new page. |
| 3. | System shows the options to add. |
| 4. | To add a worker, Section Incharge enters the worker type, name, phone no. and age in the add page. |
| 5. | Section Incharge clicks on the save button. |
| **Post-Condition** | **API** takes the request and data entered and adds the data to the database and a new worker gets added. | |
| **Alternate Flow** | NILL | |

**4.11 Use case #10**

| **UC ID** | UC#10 | |
| --- | --- | --- |
| **Use case Name** | **assign\_worker** | |
| **Description** | Section Incharge can assign work to the workers | |
| **Actor** | section\_incharge | |
| **Pre-Condition** | The Section Incharge must be logged-in.  Then he/she must click on the Unresolved complaints. | |
| **Main Flow** | 1. | System shows all the complaints done by students and staffs in the Unresolved Complaints option in the bottom left corner. |
| 2. | Section Incharge clicks on the unresolved complaints option. There he can assign a worker to a complaint by clicking on the green plus button. |
| 3. | Section Incharge assigned a complaint to a worker. |
| **Post-Condition** | Workers are assigned tasks properly to resolve problems. | |
| **Alternate Flow** | NILL | |

**4.12 Use case #11**

| **UC ID** | UC#11 | |
| --- | --- | --- |
| **Use case Name** | **removew** | |
| **Description** | Section Incharge can remove any worker from the list of workers. | |
| **Actor** | section\_incharge | |
| **Pre-Condition** | The Section Incharge must be logged-in. | |
| **Main Flow** | 1. | Section Incharge clicks on the Manage workers option in the bottom left corner. |
| 2. | Section Incharge clicks on the Workers option in the top. |
| 3. | Section Incharge selects intended worker to delete and clicks on delete button. |
| **Post-Condition** | Worker gets deleted from the list of workers. | |
| **Alternate Flow** | NILL | |

**4.13 Use case #12**

| **UC ID** | UC#12 | |
| --- | --- | --- |
| **Use case Name** | **resolvedcomplaints** | |
| **Description** | Section Incharge can view the list of resolved complaints. | |
| **Actor** | Section Incharge | |
| **Pre-Condition** | The Section Incharge must be logged-in. | |
| **Main Flow** | 1. | Section Incharge clicks on the resolved Complaints. |
| 2. | System displays a new page. |
| 3. | System shows the details of the complaints that have been resolved along with that he/she can view feedback, worker, and the caretaker associated. |
| **Post-Condition** | NILL | |
| **Alternate Flow** | NILL | |

**4.14 Use case #13**

| **UC ID** | UC#13 | |
| --- | --- | --- |
| **Use case Name** | **discharge\_worker** | |
| **Description** | Section Incharge can discharge a worker once the complaint is done resolving. | |
| **Actor** | Section Incharge | |
| **Pre-Condition** | The Section Incharge must be logged-in. Then he/she must change the status of the complaint from unresolved to resolved. | |
| **Main Flow** | 1. | Section Incharge changes the status and clicks on submit button. |
| 2. | The complainer gets notified with the new status of the work. |
| **Post-Condition** | Worker has been discharged. | |
| **Alternate Flow** | NILL | |

**4.15 Use case #14**

| **UC ID** | UC#14 | |
| --- | --- | --- |
| **Use case Name** | **caretaker\_feedback** | |
| **Description** | Caretaker can see the feedback once the complaint is done resolving. | |
| **Actor** | Care\_taker | |
| **Pre-Condition** | The caretaker must be logged-in.Then he/she must click on the resolved complaints. | |
| **Main Flow** | 1. | System displays the page of on resolved complaints where the caretaker can see all the complaints along with complainer, complaint type, date of filing, assigned worker name and feedback. |
| 2. | Caretaker clicks on the feedback button to review the feedback for a filed complaint. |
| **Post-Condition** | Caretaker can know the quality of work. | |
| **Alternate Flow** | NILL | |

**4.16 Use case #15**

| **UC ID** | UC#15 | |
| --- | --- | --- |
| **Use case Name** | **resolvepending** | |
| **Description** | Section Incharge can check the status of work assigned to the workers and all the complaints given by students and staffs. | |
| **Actor** | Section Incharge | |
| **Pre-Condition** | The Section Incharge must be logged-in. | |
| **Main Flow** | 1. | Section Incharge clicks on the Unresolved Complaints. |
| 2. | System displays a new page. |
| 3. | System shows the options for all complaints along with their complainer. |
|  | 4. | Section Incharge reviews the complaint and resolves the complaint by assigning a worker. |
| **Post-Condition** | Status of the complaint is checked by the Section Incharge | |
| **Alternate Flow** | NILL | |

**4.17 Use case #16**

| **UC ID** | UC#16 | |
| --- | --- | --- |
| **Use case Name** | **worker\_id\_know\_more** | |
| **Description** | Section Incharge can know the list of unassigned workers to assign to a complaint. | |
| **Actor** | Section Incharge | |
| **Pre-Condition** | The Section Incharge must be logged-in.Then he/she must click on the unresolved complaints thereafter on Pending complaints. | |
| **Main Flow** | 1. | System displays the page of on pending where the caretaker can see all the complaints. |
| 2. | Section Incharge clicks on the green button at the corner. |
| 3. | System displays a list of unassigned workers who are ready to work. |
| **Post-Condition** | Section Incharge knows the list of unassigned workers. | |
| **Alternate Flow** | NILL | |

**4.18 Use case #17**

| **UC ID** | UC#17 | |
| --- | --- | --- |
| **Use case Name** | **changestatus** | |
| **Description** | Caretaker can change the status of the complaint. | |
| **Actor** | Caretaker | |
| **Pre-Condition** | The Caretaker must be logged-in.Then he/she must click on the unresolved complaints thereafter on On Hold Complaints. | |
| **Main Flow** | 1. | System displays the page of on hold complaints. |
| 2. | Caretaker clicks on the change status button at the corner. |
| 3. | Caretaker changes the state of work from unresolved to resolved. |
| **Post-Condition** | Student/Staff gets notified with the changed status of the complaint. | |
| **Alternate Flow** | NILL | |

**5. NON-FUNCTIONAL REQUIREMENTS**

**NR0. Ensuring security**

The user has to login before he/she could access anything or make a complaint. Admin can only make changes in the categories of complaints and other related issues. Access to any application resource will depend upon user’s designation.

**NR1. Performance Requirements**

The server on which the website and database are hosted must be online 24X7.It must be able to handle multiple requests of different users at a time.

Response time of the complaint management system should be less. Response time refers to the waiting time while the system accesses queries and retrieves the information from the database**.**

**NR2. Reliability**

It should be robust with high degree of fault tolerance. Any user error should be identified and user should be notified about it. It should have mechanisms to overcome hardware failures, power failures and roll back databases to their last checkpoint. This application should be highly reliable and it should generate all the updated information by student, faculty and administrator in correct order.

**NR.3 Usability**

The interface should be easy to use. The web interface should be intuitive and easy to navigate. Users should be able to understand the menu and options provided by the software. Any notification or error messages should be precise and clear. The software should be able to understand and capable to learn, Use and attractive to the user when used under specified condition.

**NR4. Integrity**

Only system administrator should have right to change system parameters. The database should be protected.