Software Requirements & Specification

Semester Project-Fall 2022

BS(SE) Semester 5

IMSciences, Peshawar Course Instructor:

Hamood Ur Rehman Durrani

Assistant Professor

(Software Engineering)

IMSciences Peshawar

Contents

[1.1 Introduction 3](#_Toc123774924)

[1.2 Project Overview 3](#_Toc123774925)

[1.3 Problem Statement 4](#_Toc123774926)

[1.4 Objectives 4](#_Toc123774927)

[1.5 Existing System 4](#_Toc123774928)

[2. Functional Objectives 4](#_Toc123774929)

[2.1 High Priority 4](#_Toc123774930)

[2.2 Medium Priority 5](#_Toc123774931)

[2.3 Low Priority 5](#_Toc123774932)

[3. Non-Functional Objectives 6](#_Toc123774933)

[3.1 Reliability 6](#_Toc123774934)

[3.2 Usability 6](#_Toc123774935)

[3.3 Performance 6](#_Toc123774936)

[3.4 Security 6](#_Toc123774937)

[3.5 Online user Documentation and Help 6](#_Toc123774938)

[3.6 Maintainability 7](#_Toc123774939)

[3.7 Interfaces 7](#_Toc123774940)

[3.8 Portability 7](#_Toc123774941)

[3.9 Legislative 7](#_Toc123774942)

[4. The Use Case Model 8](#_Toc123774943)

[4.1 System Use Case Diagram 9](#_Toc123774944)

**Team Name:** E. A. H

**Team Members:** Enayatullah, Anas Bin Zia, Hebatullah

**Hostel Management System**

## **1.1 Introduction**

Most of hostels don’t have management systems. So, do their works manually and mostly their documents are handwritten which is not a wise job. Thus, many repetitions and difficulties can be overcome with an automated system. A computerized system can reduce a lot of manual works, improve the efficiency of the system, and overcome many drawbacks of existing system.

This system is designed to manage hostel activities including room allocation and deallocation, update staffs and boarders’ information, calculating hostel fees, staffs’ salary, furniture information, check-in check-out premises, get feedback based on the room, food, and other amenities. Boarders also can check online whether the hostel has or not

empty rooms, its fees, mess menu, number of people staying in one room, get admission and allocate room for themselves in better way. The administration has a unique identity for each member as well as student’s details.

The user community includes students, faculty members, hostel administration, hostel staffs, university admission office, university accounting office, transportation.

## **1.2 Project Overview**

To provide university students apartment to the university hostel more efficiently, this web-based software is designed. This project also saves information’s of hostellers and those who left the hostel in previous years. This project is going to minimize human works and hostel registration easier for students by providing online application. It is headed by warden. He will be the administrator. Students can be informed through notice board, hostel fee and mess menu by login into the online system.

## **1.3 Problem Statement**

There exists a lot of drawbacks in keeping and updating a hostel information. Especially if the system is manual. Most of hostels are being run by one warden, the number of students staying in each room is not known for the officer. In absence of the warden, the hostel officer needs to go and check each room to check if there is a free set. Or to find a proper student he needs to check all the papers to find him/her. So, it will be very difficult and lengthy.

## **1.4 Objectives**

* To make the hostel management easier, more efficient.
* To maintain and collect students’ information easily and properly.

## **1.5 Existing System**

The existing system is manual and there many drawbacks and time consumption activities are done. In the existing system it is possible to apply for hostel online but the allocation processes are done manually which may lead problems in allocation process.

# **2. Functional Requirement**

## **2.1 High Priority**

1. The system shall help administrator to keep and update hostellers’ information online. This feature will prevent form missing information, repeating same processes, using papers, and reduce time and cost.
2. The system shall help students to get registration in hostel, and allocate or deallocate room online.
3. The system shall help the security guard to know whom should be allowed to enter in hostel and whom shouldn’t. This feature helps security guard to have more control on security. And also, non-hostellers will not be able to use hostel facilities.
4. The system shall help the administrator to update the information of those who are deallocating rooms.
5. The system shall update information of those paid their fees and those who haven’t. This feature helps the administrator to know whom should inform to release his/her dues.

## **2.2 Medium Priority**

1. The system shall display information of the furniture and facilities that the hostel will provide for hostellers. This feature will make users more confident whether to register in hostel or not.
2. The system shall allow the staff to check their attendance, working hours, working area, activities, salary online.
3. The system shall allow students to give their feedback about the hostel facilities and services. This feature helps the administrator to know in which services and facilities should bring change and what problems should be solved to keep hostellers satisfied.
4. The system shall allow health care regulators to have supervision on hostel’s food items.

## **2.3 Low Priority**

1. The system shall allow the new user to sign up in the system by entering his/her email address, password, residential address such as country, city, street name, house number, age, gender, contact number.
2. The system shall announce the updates in notice board. This feature helps the hostellers to know about changes and updates in hostel. And it will help the administrator to inform all the hostellers about the changes and updates.
3. The system shall connect to the university database of rule and regulation for check in hostel.

# **3. Non-Functional Objectives**

## **3.1 Reliability**

• The system shall be completely operational at least 80% of the time.

• Down time after a failure shall not exceed 20 minutes.

## **3.2 Usability**

• The administrator should be able to use the system in his job after days of

training.

• A user who already knows what information he is interested in should be able to

view that page in 3 seconds.

## **3.3 Performance**

• The system should be able to support 100 simultaneous users.

• The average time that the web pages take to be opened in 30kbps internet connection should be maximum of 2 seconds.

* The system should be able to store at least 1000 users’ information in its database.
* The average time to upload record of users in maximum of 30 seconds.
* The system should be able to support 200 concurrent users at same time.

## **3.4 Security**

• The system shall provide password protected access to web pages that are to be

viewed only by users.

* The system should not allow unauthorized users to access.
* The system should not lose its data when it fails.

## **3.5 Online user Documentation and Help**

• The system shall provide a web page that explains how to navigate the site.

This page should be customized based on what pages that user is allowed to

access.

• This help page should be accessible from all other pages.

## **3.6 Maintainability**

* The system should be able to be updated.
* The system should support the future modification.

## **3.7 Interoperability**

The system must interface with

• The current Oracle database systems for services and facilities information

• The current Oracle Financial accounting system

• The acquired language translation tool

• The acquired web site search engine

## **3.8 Portability**

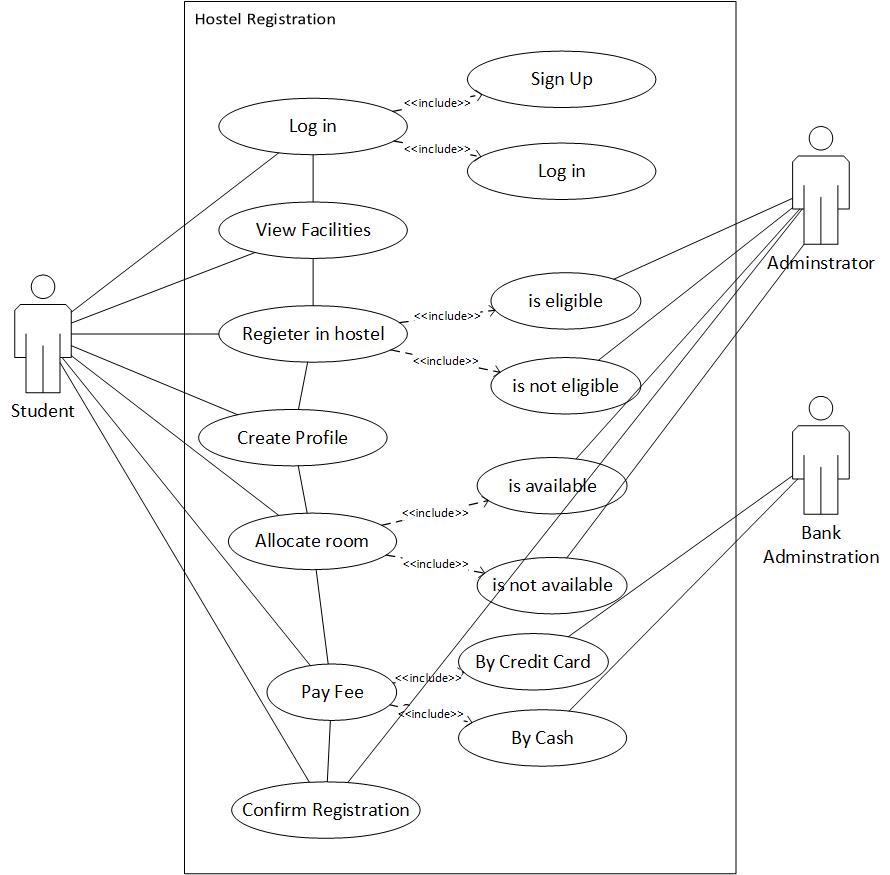
* The system should be runnable on Linux, Windows, Mac operating system.
* The system should be applicable on Google chrome, fire fox, internet explorer, Microsoft edge, safari, opera browser.

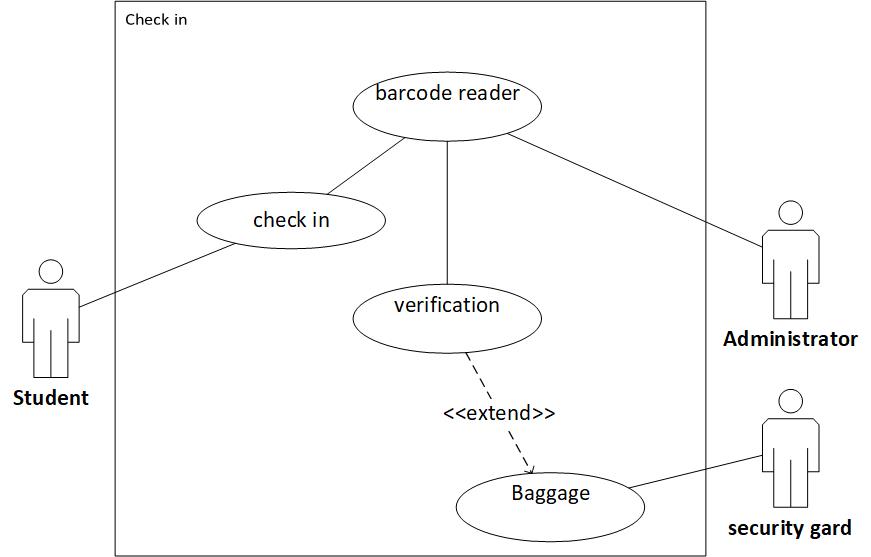
## **3.9 Legislative**

* The system should follow the rules of Research of National Assembly of Pakistan.

# **4. The Use Case Model**

**4.1 Use Case Diagram**





## **4.1 System Use Case Diagram Description**

**Login User**

|  |  |
| --- | --- |
| **Use Case Name** | **Log in** |
| **Summery** | In order to view the facilities, register in a hostel, check details the user must log in first in the system. |
| **Basic Flow** | 1. The user wants to log in. 2. The system requests for user name and password. 3. The user enters his/her username and password. 4. The system verifies his/her username and password. 5. The user logs in. |
| **Alternative Flows** | **Step 4:**  The username or password is incorrect.  **Step 4:**  The user enters his username or/and password incorrectly. The system shows a message that inform the user to repeat the step 2. If the user enters his/her username and password correctly, the system will continue the process from step 4. |
| **Pre-Condition** | The user should have created account in the system already. |
| **Post Condition** | The user now can have accessed the data according to his registered access level. |
| **Business Rule** | Different users can access different levels according to their registered access level. Such that students access level will be different to staff and wardens. |

**Sign up User**

|  |  |
| --- | --- |
| **Use Case Name** | **Sign Up** |
| **Summery** | In order to be able to use the system, the user should sign up in the system first. |
| **Basic Flow** | 1. The user wants to sign up. 2. The system requests to enter user name, password, age, gender, residential address, phone contact information. 3. The user enters the information. 4. The system checks that the user doesn’t duplicate an existing account. 5. The system verifies the user. 6. The user signs up in the system. |
| **Alternative Flows** | **Step 4:**  The user duplicates and existing user account.  **Step 4:**  The user enters the username of an existing account. The system shows a message that it is signed up already. The system will request the user to repeat step 3. |
| **Pre-Condition** | The user should not have an existing account in the system. |
| **Post Condition** | The user now has signed up and can log in the system. |
| **Business Rule** | none |

|  |  |
| --- | --- |
| **Use Case Name** | **Check Details** |
| **Summery** | In order to check their working area, jobs, working hours, attendance, salary the staffs should check the system. |
| **Basic Flow** | 1. The user wants to choose the information that he/she wants to check. Such working area, working hours, attendance, salary. 2. The system shows the asking information. |
| **Alternative Flows** | none |
| **Pre-Condition** | The user should be a staff member of the hostel and logged in. |
| **Post Condition** | The user now can have accessed the information that he/she wants to know and check his/her attendance. |
| **Business Rule** | Different users can access different levels according to their registered access level. |

**Check Details (Staff)**

**Checking Quality (Furniture)**

|  |  |
| --- | --- |
| **Use Case Name** | **Checking Quality** |
| **Summery** | In order to view the facilities, and furniture’s quality the user must have access the system. |
| **Basic Flow** | 1. The user wants to check the furniture available in hostel. 2. The system lists all the available furniture. 3. The user wants to check the quality of a furniture such as chairs, desks, beds, cabinets. The user will click on the particular furniture name. 4. The system shows the pictures of the chosen furniture. |
| **Alternative Flows** | **Step 1:**  It is not available in the hostel.  **Step 1:**  The user search for a furniture that is not available in hostel. The system will show a message that the searched item is not available. The user will repeat step 1. |
| **Pre-Condition** | The user should have created account and logged in the system already. |
| **Post Condition** | The user now can check the furniture and their qualities that are available in hostel. |
| **Business Rule** | Different users can access different levels according to their registered access level. |

|  |  |
| --- | --- |
| **Use Case Name** | **View Facilities** |
| **Summary** | This use case allows a registered user to view the facilities that the hostel provides to its boarders. |
| **Basic Flow** | 1. The use case starts when a user wants to view the facilities. 2. The system provides list of facilities of the hostel. 3. The user wants to see if the desired facility is available 4. The system provides the picture of the facility. 5. The user wants to check the menu. 6. The current menu will appear. |
| **Alternate Flow** | **Step 3:**  The facility the user is searching is not available.  **Step 3:**  The user searches for gym, it is not available, the system will show a message that this facility is not available. The user will repeat step 3. |
| **Preconditions** | The user has an account and is logged in. |
| **Post conditions** | The user can view the facilities. |
| **Business rules** | Different users can access different levels according to their registered access level. Such that students access level will be different to staff and wardens. |

**View Facilities**

**Register**

|  |  |
| --- | --- |
| **Use Case Name** | **Register** |
| **Summery** | The user wants to get registration in hostel. The system will provide the steps to get registration. |
| **Basic Flow** | 1. The user wants to get registration. 2. The system requests the user to create his/her profile and enter the information needed. Such that photo, full name, father’s name, degree name, semester, contact information. 3. The user adds the information. 4. The system confirms the information. 5. The system provides payment methods. Either through cash or credit card. 6. The user chooses a payment method and does the payment. 7. The system requests for pdf copy of payment transcript. 8. The user uploads the fee payment transcript pdf copy. 9. The system confirms the registration. |
| **Alternate Flow** | **Step 2:**  Duplicated information  **Step 2:**  The user enters the existing information. The system shows a message that the entered information exists already. The user repeats step 2.  **Step 6:**  Insufficient balance.  **Step 6:**  The user’s balance is not enough. The system informs the user to recharge his/her balance and repeat step 5. |
| **Preconditions** | The user has an account, is logged in and has checked the facilities |
| **Postconditions** | The user is now a boarder of the hostel. |
| **Business rules** | Different users can access different levels according to their registered access level. Such that students access level will be different to staff and wardens. |

**Check in Student**

|  |  |
| --- | --- |
| **Use case name:** | **Check in student** |
| **Summery:** | In order for student to enter in hostel, the security authority through barcode reader will check his/her identity and material he/she carries to avoid violation of University regulation. |
| **Normal flows:** | 1. The student wants to enter hostel. 2. The system barcode reader asks for student – card 3. The student touches his/her card to barcode reader system. 4. The system verifies the student information and guarantee the permission to student via message in board written “hostellers “. 5. The guard carrying the system check the baggage of student for security issues and verify via a “sound” 6. The student enters to hostel. |
| **Alternative flows:** | **Step 2:**  student forget his/her student card  **Step3:**  student card denied by system suggest to go to security department  **Step 5:**  student baggage has security issues suggest to investigate in details and do step 5 again. |
| **Extension point:** | None |
| **Precondition:** | The student should be registered in university and hostel |
| **Post condition:** | The student now can go to his/her room and use the facilities of hostel according to his/her register level. |
| **Business Rules:** | Students are not allowed to go to cooking room. |

**Give feedback**

|  |  |
| --- | --- |
| **Use case name:** | **Give feedback** |
| **Summery** | In order to increase the hostel’s quality ad furniture , we need students satisfaction feedback based on food, room and amenities though entering his/her account email feedback to quality assurance staff of hostel. |
| **Normal flows:** | 1. Students login to his/her account 2. The system asks for username and password 3. The student enters his/her username and password. 4. After authentication, student click on option of feedback 5. The system displays feedback menus 6. The student marks the quality of hostel furniture, food, room, and amenities through wizards “low”,” high”,” medium” and at the end suggestion of student. 7. The student submits his/her feedback. 8. The system display a message of confirmation. |
| **Alternative flows:** | **Step 3:**  if username invalid goes back to step 2  If password invalid the system request to re-enter the password and continue with step 4  **Step 7:**  if student did not check any option of feedback form the system request to fill the form from step 6. |
| **Extension point:** | None |
| **Precondition:** | The student should be registered in hostel. |
| **Post condition:** | The quality assurance now can determine which food item or furniture to boost in order to get satisfactions of student. |
| **Business rules:** | Some material of food item is regulated by external regulation like health-care therefore it is unchangeable. |

**5 Requirement Analysis**

**5.1 Quality Attribute Table**

|  |  |  |
| --- | --- | --- |
| **Requirement Attribute** | **Example of bad requirement** | **Example of good requirement** |
| **Atomic** | The system verifies the student information and guarantee the permission to student via message in board written “hostellers “. | The system verifies the student information.  The system guarantees the permission to student via message “host-elite” |
| **Uniquely identified** | 1. The system shall help administrator to keep and update hostellers’ information online. This feature will prevent form missing information, repeating same processes, using papers, and reduce time and cost. | 1. The system shall help administrator to keep the hosteller data 2. The system shall help administrator to update hosteller’ data online. |
| **Complete** | The administrator should be able to use the system in his job after days of training. | The administrator should be able to use the system in his/her  Job after three days of training |
| **Consistence and unambiguous** | The system should be able to support 100 simultaneous users.  The system should be able to support 200 concurrent users at same time. | The system should able to support 100 simultaneous users. |
| **Traceable** | Maintain staff information mapped to FR req.ID? | Maintain staff information mapped to FR req.ID 2.2 |
| **Prioritized** | The system shall connect to the university database of rule and regulation for check in hostel. (low-priority)  The system shall allow health care regulators to have supervision on hostel’s food items. (medium priority) | The system shall allow health care regulators to have supervision on hostel’s food items. (high priority)  The system shall connect to the university database of rule and regulation for check in hostel. (medium-priority) |
| **Testable** | The system should be able to be updated | The system should be able to be update after a period of two months. |

**5.2 Requirement Checklist -1**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Easy to learned | Easy to understand | Easy to operate |
| The administrator should be able to use the system in his job after two days of training. | ✓ | ✓ | ✓ |
| A user who already knows what information he is interested in should be able to  view that page in 3 seconds | ✓ | ✓ | ✓ |
| The system should be able to store at least 1000 users’ information in its database. | ✓ | ✓ | ✓ |
| The system shall be completely operational at least 80% of the time. | ✓ | ✓ | ✓ |

**5.3 Positive and negative among selected quality attributes**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Reliability | Usability | Performance | Security | Maintainability | Interoperability | Portability | Legislative |  |  |  |  |  |  |  |  |  |
| Reliability |  | **+** | ­­­­­­­- | + | + |  |  | + |  |  |  |  |  |  |  |  |  |
| Usability | + |  | - |  | - |  | - | + |  |  |  |  |  |  |  |  |  |
| Performance |  | - |  |  |  |  | - |  |  |  |  |  |  |  |  |  |  |
| Security | + | - | - |  | - |  | - | + |  |  |  |  |  |  |  |  |  |
| Maintainability | + | + |  | + |  |  |  | + |  |  |  |  |  |  |  |  |  |
| Interoperability | + |  | - | - |  |  | + | - |  |  |  |  |  |  |  |  |  |
| Portability |  | - | - | - | + | + |  |  |  |  |  |  |  |  |  |  |  |
| Legislative |  | - | - | + | - | + |  |  |  |  |  |  |  |  |  |  |  |

**5.4 quality attribute prioritization**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Performance | Reliability | Security | Usability | Maintainability | Interoperability | Portability | Legislative |
| Performance | **4** |  | **<** | **^** | **^** | **<** | **<** | **<** | **^** |
| Reliability | **2** |  |  | **^** | **^** | **^** | **<** | **<** | **^** |
| Security | **7** |  |  |  | **<** | **<** | **<** | **<** | **<** |
| Usability | **5** |  |  |  |  | **<** | **<** | **<** | **^** |
| Maintainability | **1** |  |  |  |  |  | **^** | **^** | **^** |
| Interoperability | **1** |  |  |  |  |  |  |  | **^** |
| Portability | **1** |  |  |  |  |  |  |  | **^** |
| Legislative | **6** |  |  |  |  |  |  |  |  |

**6 Requirements Validation**

**6.1 Requirements Checklist**

|  |  |  |
| --- | --- | --- |
| Checklist Attribute | Bad Example | Good Example |
| Understandability | The system shall allow students to give their feedback about the hostel facilities and services. This feature helps the administrator to know in which services and facilities should bring change and what problems should be solved to keep hostellers satisfied | The “satisfaction factor “ was not defined in glossary of requirements documents, therefore the reader cannot understand the purpose of author |
| Redundancy | The system shall help the administrator to update the information of those who are deallocating rooms. | This service was repeated in many requirements, in 1- high priority requirements 2- in medium priority requirements. |
| Completeness | The system shall display information of the furniture and facilities that the hostel will provide for hostellers. This feature will make users more confident whether to register in hostel or not. | There is missing requirement here, where the system should display the furniture and facilities’ information. |
| Ambiguity | The system should be able to support 200 concurrent users at same time. | The author uses the term concurrent and simultaneous and it’s the place of misinterpretation. |
| Consistency | Security | The non-functional security has contradiction with all non-functional requirements |
| Organization | Reliability: The system should not lose its data when it fails.  Not grouped to right requirement | Security : The system should not lose its data when it fails. |
| Conformance to Standard | The average time that the web pages take to be opened in 30kbps internet connection should be maximum of 2 seconds. | This requirement violates the standard of internet the expert did not deliver the domain knowledge properly. |
| Traceability | Maintain staff information mapped to FR req.ID? | Maintain staff information mapped to FR req.ID 2.2 |

**6.2 Test Case**

|  |  |
| --- | --- |
| **Test Case Field** | **Description** |
| **Test case ID:** | “TC\_FR\_1” |
| **Test Priority:** | * High |
| **Name of the Module**: | Functional Requirements |
| **Test Designed by**: | Enayatullah |
| **Date of test designed**: | 2023/1/6 |
| **Test Executed by**: | Enayatullah |
| **Date of the Test Execution**: | 2023/1/6 |
| **Name or Test Title**: | Check in student |
| **Description/Summary of Test**: | In order for student to enter in hostel, the security authority through barcode reader will check his/her identity and material he/she carries to avoid violation of University regulation. |
| **Pre-condition**: | The student should be registered in university and hostel |
| **Dependencies**: | This test case depends on register test case |
| **Test Steps**: | 1. The student comes in front of hotel to enter it. 2. The barcode reader asks for student-card. 3. The students should touch his/her card to barcode reader system for validation. 4. The system checks the student card according to a predefine policy provided to the system and confirm via a message displayed “hostellers” 5. The guard carrying a system check the students’ baggage for security issues and verify through a “sound” message. 6. After confirmation of all above step student will enter to hostel. |
| **Test Data**: | Enayat-card, Anas-card, Hebat-card |
| **Expected Results**: | 1. Confirm entrance when all test-steps are correct 2. Invalid student card and unrelated message like “not hostilities” will appear in screen 3. A warning sound alert when baggage has security issues. |
| **Post-Condition**: | The student can enter to hostel and use hostel’s facilities |
| **Actual Result**: | The barcode system and security guard granted students entrance |
| **Status (Fail/Pass):** | “Pass” |
| **Notes**: | The student should carry his/her student-card |

|  |  |
| --- | --- |
| **Test Case Field** | **Description** |
| **Test case ID:** | “TC\_FR\_2” |
| **Test Priority:** | * Medium |
| **Name of the Module**: | Functional Requirements |
| **Test Designed by**: | Enayatullah |
| **Date of test designed**: | 2023/1/6 |
| **Test Executed by**: | Enayatullah |
| **Date of the Test Execution**: | 2023/1/6 |
| **Name or Test Title**: | Give Feedback |
| **Description/Summary of Test**: | In order to increase the hostel’s quality ad furniture , we need students satisfaction feedback based on food, room and amenities though entering his/her account email feedback to quality assurance staff of hostel. |
| **Pre-condition**: | The student should be registered in university and hostel |
| **Dependencies**: | This test case depends on register test case |
| **Test Steps**: | 1. The student should login to his/her account. 2. System ask for username and password 3. The student enters his/her username and password. 4. System authenticate the students on basis of predefined policy for logging. 5. The student clicks on option of feedback. 6. The system displays feedback menus on set of predefined values. 7. The system checks the quality of hostel furniture, food, room, and amenities through values “low”,” high”,” medium” and at the end suggestion of student on basis of student’s choice. 8. The system displays confirmation message on basis of students submit |
| **Test Data**: | 1. Food-item, 2- furniture, 3- room structure |
| **Expected Results**: | Best food item quality furniture |
| **Post-Condition**: | The quality assurance now can determine which food item or furniture to boost in order to get satisfactions of student. |
| **Actual Result**: | System did not respond properly on menu, which provided for checking food item and furniture, therefor quality assurance cannot see the student’s feedback on food and furniture. |
| **Status (Fail/Pass):** | “Fail” |
| **Notes**: | The student should carry his/her student-card |

**6.3 Paraphrasing**

Now we should change our system requirement from the natural language to some System model to paraphrase the changes that occurred in our system requirement and compare them for validity or simply highlighting the changes that we made up to now for validation, I have done those in original functional and non-functional requirements therefore I do not repeat again.