COMSATS University Islamabad, Abbottabad Campus

Department of Computer Science

CSC392 Object-Oriented Software Engineering

**Project Proposal**

**HOSTEL MANAGEMENT SYSTEM**

|  |
| --- |
| **Group Members** |
| FA21-BSE-015  Hanzla Nouman |
| FA21-BSE-019  Laiba Binta Tahir |
| FA21-BSE-027  Irfan |
| FA21-BSE-080  Arfah Ali |

…

Table of Contents

[CHAPTER 1 PROJECT PROPOSAL 3](#_Toc131566468)

[Introduction 3](#_Toc131566469)

[We can improve the efficiency of the system, thus overcome the drawbacks of the existing system. 3](#_Toc131566470)

[Vision and Business Case 4](#_Toc131566471)

[Use-Case Model 4](#_Toc131566472)

[Functional Requirements for Hotel Management System 4](#_Toc131566473)

[Supplementary Specification 5](#_Toc131566474)

[Glossary 6](#_Toc131566475)

[Risk List & Risk Management Plan 6](#_Toc131566476)

[CHAPTER 2: USE CASES 7](#_Toc131566477)

[Use Case Diagram 7](#_Toc131566478)

[Use Cases Distribution 8](#_Toc131566479)

[Brief Level Use Cases 10](#_Toc131566480)

[Student Name 1 (Registration Number 1) 10](#_Toc131566481)

[Use Case: Process Sale 10](#_Toc131566482)

[Fully Dressed Use Cases 11](#_Toc131566483)

[Student Name 1 (Registration Number 1) 11](#_Toc131566484)

[Use Case UC1: Process Sale 11](#_Toc131566485)

# CHAPTER 1 PROJECT PROPOSAL

## Introduction

1.1 PROJECT OVERVIEW

The hostel management system is web-based software to provide university students accommodation to the university hostel more efficiently. This project also keeps details of the hostellers and applied students. It is headed by multiple administrators . This document is intended to minimize human works and make hostel allocation an easier job for students and hostel authorities by providing online application for hostel, automatically select the students from the waiting list and mess calculation, complaint registration, etc. Students will get approval notification in their mails. Hostellers can view hostel fee and mess menu by login into the online system.

1.2 PROBLEM STATEMENT

There are a lot of drawbacks in keeping and maintaining a hostel. Especially with a manual system. Since most hostels are being run by only one hostel manager, the number of students in a room are sometimes not known by the warden.

This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented.

# We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

1.3 OBJECTIVES

• To make it easier for data collection, storage and referencing reliable.

• To maintain the students as hostellers and waiting list students separately.

• To process allotment list.

## Vision and Business Case

*Describes the high-level goals and constraints, the business case, and provides an executive summary.*

[Text Book: 7.6. NextGen Example: (Partial) Vision]

## Use-Case Model

# Functional Requirements for Hostel Management System

1. Allocaton /Booking

* record reservations
* record the student first name
* record the student’s last name
* record the number of students
* record the room number
* display the default room fee display whether or not the room is guaranteed.
* generate a unique confirmation number for each.
* record the expected check-in date and time
* The system shall record the expected checkout date and time
* The system shall record customer feedback

2.  Food

* The system shall track all meals purchased in the hotel .
* The system shall record payment and payment type for meals
* The system shall bill the current room if payment is not made at time of service
* The system shall accept reservations for services.

3. Management

* display the hotel occupancy for a specified period of time (days; including past, present, and future dates).
* display projected occupancy for a period of time (days).
* display room revenue for a specified period of time (days).
* display food revenue for a specified period of time (days).
* display an exception report, showing where default room and food prices have been overridden
* allow for the addition of information, regarding rooms, rates, menu items, prices, and user profiles
* allow for the deletion of information, regarding rooms, rates, menu items, prices, and user profiles
* allow for the modification of information, regarding rooms, rates, menu items, prices, and user profiles
* allow managers to assign user passwords

## Supplementary Specification

There are a lot of software requirements specifications included in the non-functional requirements of the Hostel Management System, which contains various processes, namely Security, Performance, Maintainability, and Reliability.

**Security:**

● user Identification: The system needs the user to recognize herself or himself using the phone.

● Logon ID: Any users who make use of the system need to hold a Logon ID and password.

● Modifications: Any modifications like insert, delete, update, etc. for the database can be synchronized quickly and executed only by the ward administrator.

● Front Desk Staff Rights: The staff at the front desk can view any data in the Hostel Management system.

● Administrator rights: The administrator can view as well as alter any information in the Hostel Management System.

**Performance:**

● Response Time: The system provides acknowledgment in just one second once the 'user's information is checked.

● Capacity: The system needs to support at least 1000 people at once.

● User-Interface: The user interface acknowledges within five seconds.

● Conformity: The system needs to ensure that the guidelines of the Microsoft accessibilities are followed.

**Maintainability:**

● Back-Up: The system offers efficiency for data backup.

● Errors: The system will track every mistake as well as keep a log of it.

**Reliability:**

● Availability: The system is available all the time.

## Glossary

|  |  |
| --- | --- |
| *HMS* | Hostel management system |
| *DFD* | Data flow diagram |
|  |  |
|  |  |

## 

## Risk List & Risk Management Plan

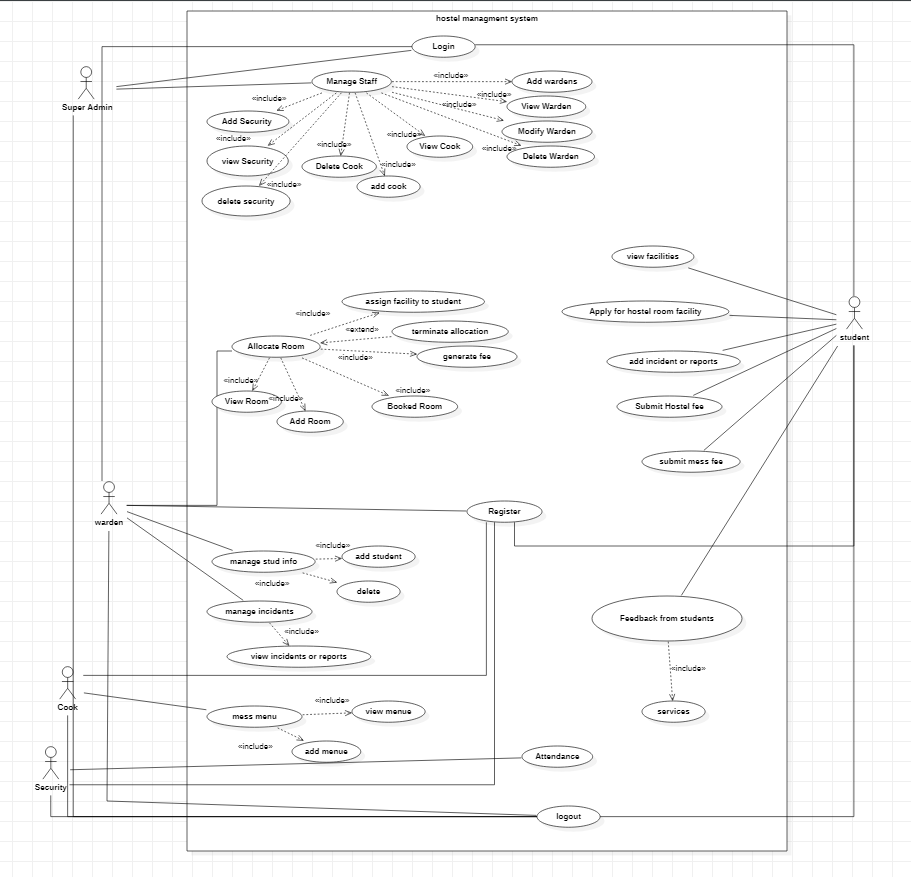
*Describes the risks (business, technical, resource, schedule) and ideas for their mitigation or response.*

* + For example, leadership really wants a demo at the POSWorld trade show in Hamburg, in 18 months. But the effort for a demo cannot yet be even roughly estimated until deeper investigation.
  + For example, in the NextGen POS, for what remote services (e.g., tax calculator) will fail-over to local services be allowed? Why? Do they provide exactly the same services locally as remotely, or are there differences?

# 

# CHAPTER 2: USE CASES

## Use Case Diagram



## Use Cases Distribution

|  |  |  |
| --- | --- | --- |
| S#. | Group Member | Assigned Use Cases |
| 1 | FA21-BSE-015  Hanzla Nouman | Allocate rooms:   * Add room * View room * Generate fee * Terminate allocation * Assign facilities to students * Booked room |
| 2 | FA21-BSE-019  Laiba binta tahir | Feedback from students  Mess menu  View incidents and reports.  Attendance  Logout  Manage students.   * Add student * View student * Delete student * modify student |
| 3 | FA21-BSE-027  Irfan | Login  Register  Manage warden   * Add warden * View warden * Delete warden * modify warden   Manage staff   * Add security * View security * Delete security * Add cook * View cook * Delete cook |
| 4 | FA21-BSE-080  Arfah Ali | View facilities  Apply for hostel  Add incidents  Submit fee(hostel/mess)  View incidents |

## Brief Level Use Cases

### Student Name 1 (Registration Number 1)

#### Use Case: Process Sale

A customer arrives at a checkout with items to purchase. The cashier uses the POS system to record each purchased item. The system presents a running total and line-item details. The customer enters payment information, which the system validates and records. The system updates inventory. The customer receives a receipt from the system and then leaves with the items.

Student Name 2 (Registration Number 2)

Student Name 3 (Registration Number 3)

## Fully Dressed Use Cases

### Student Name 1 (Registration Number 1)

| Use Case UC1: Process Sale |
| --- |
| Scope: NextGen POS application  Level: user goal  Primary Actor: Cashier  Stakeholders and Interests:  - Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her salary.  - Salesperson: Wants sales commissions updated.  - Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of entered items and prices. Wants proof of purchase to support returns.  - Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded. Wants some fault tolerance to allow sales capture even if server components (e.g., remote credit validation) are unavailable. Wants automatic and fast update of accounting and inventory.  - Manager: Wants to be able to quickly perform override operations, and easily debug Cashier problems.  - Government Tax Agencies: Want to collect tax from every sale. May be multiple agencies, such as national, state, and county.  - Payment Authorization Service: Wants to receive digital authorization requests in the correct format and protocol. Wants to accurately account for their payables to the store.  Preconditions: Cashier is identified and authenticated. |

Success Guarantee (or Postconditions): Sale is saved. Tax is correctly calculated. Accounting and Inventory are updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded.

Main Success Scenario (or Basic Flow):

1. Customer arrives at POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules.

Cashier repeats steps 3-4 until indicates done.

1. System presents total with taxes calculated.
2. Cashier tells Customer the total, and asks for payment.
3. Customer pays and System handles payment.
4. System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
5. System presents receipt.
6. Customer leaves with receipt and goods (if any).

Extensions (or Alternative Flows):

\*a. At any time, Manager requests an override operation:

1. System enters Manager-authorized mode.
2. Manager or Cashier performs one Manager-mode operation. e.g., cash balance change, resume a suspended sale on another register, void a sale, etc.
3. System reverts to Cashier-authorized mode.

\*b. At any time, System fails:

To support recovery and correct accounting, ensure all transaction sensitive state and events can be recovered from any step of the scenario.

1. Cashier restarts System, logs in, and requests recovery of prior state.
2. System reconstructs prior state.

2a. System detects anomalies preventing recovery:

* 1. System signals error to the Cashier, records the error, and enters a clean state.
  2. Cashier starts a new sale.

1a. Customer or Manager indicate to resume a suspended sale.

1. Cashier performs resume operation, and enters the ID to retrieve the sale.
2. System displays the state of the resumed sale, with subtotal.

2a. Sale not found.

* 1. System signals error to the Cashier.
  2. Cashier probably starts new sale and re-enters all items.

1. Cashier continues with sale (probably entering more items or handling payment).

Special Requirements:

- Touch screen UI on a large flat panel monitor. Text must be visible from 1 meter.

- Credit authorization response within 30 seconds 90% of the time.

- Somehow, we want robust recovery when access to remote services such the inventory system is failing.

- Language internationalization on the text displayed.

- Pluggable business rules to be insertable at steps 3 and 7.

- …

Technology and Data Variations List:

\*a. Manager override entered by swiping an override card through a card reader, or entering an authorization code via the keyboard.

3a. Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.

3b. Item identifier may be any UPC, EAN, JAN, or SKU coding scheme.

7a. Credit account information entered by card reader or keyboard.

7b. Credit payment signature captured on paper receipt. But within two years, we predict many customers will want digital signature capture.

Frequency of Occurrence: Could be nearly continuous.

Open Issues:

- What are the tax law variations?

- Explore the remote service recovery issue.

- What customization is needed for different businesses?

- Must a cashier take their cash drawer when they log out?

- Can the customer directly use the card reader, or does the cashier have to do it?

Screen Shots:

<Paste the images from netbeans here for the relevant screen>