Hall Management System

Submitted to:  
**Sajjadur Rahman**  
Lecturer, Dept. of CSE

**Johra Muhammad Moosa**Lecturer, Dept. of CSE

Submitted by:

Shishir Kumar Sarkar (1005040)  
Yahya Ahmed Sharif (1005048)  
Ehsan-Ul-Haque (1005052)  
Mahmood Ahmed (1005055)  
Nayeem Islam (1005060)

Introduction:  
 Our present hall management system is way too backdated. This is not suitable for both the students and the hall management authority. To keep up with this current digital world we need an upgraded well design information system.

Subsystems:

1. Computerized Hall Seat Allocation

2. Hall Mess management System

3. Virtual Notice Board & Complain box

4. Room Maintenance System

5. Event Management System

SUBSYSTEM 1

COMPUTERIZED HALL SEAT ALLOCATION

There are seven use cases

* Apply for seat
* Eligibility checking
* Approval
* Check for seat
* Seat allocation
* Inform student
* Inform authority

1.1APPLY FOR SEAT

Figure 1.1.1 Activity diagram for apply for seat

In this figure, we see that the steps a student has to follow to submit a application form for the hall seat. This process is finished when the application is received by the hall authority.

1.1APPLY FOR SEAT



Figure 1.1.2 Sequence diagram for apply for seat

In this diagram we see that the student fills an application in the ApplicationUI class and after that this class sends the application using the sendApplication() method. The application is then stored in the ApplicationRecieved class.

1.1APPLY FOR SEAT



Figure 1.1.3 State diagram for apply for seat

In this figure, we see that the entity class ApplicationReceiver sends a request for application and if there is any application, it receives the application and the process comes to an end.

1.2 ELIGIBILITY CHECKING

Fig 1.2.1 Activity diagram for eligibility checking

In this figure, we see that the steps of activities in eligibility checking. When the provost gets notified, a meeting is called between him and the student. After the meeting a decision is made.

1.2 ELIGIBILITY CHECKING



Fig 1.2.2 Sequence diagram for eligibility checking

In this diagram we see that the provost accepts the application and then a meeting is arranged. The control class Interview sends a notification to the student. After the meeting, the decision is made is saved by the saveDecision() method.

1.2 ELIGIBILITY CHECKING



Fig 1.2.3 State diagram for eligibility checking

In this figure, we see that the entity class Decision checks application request and if the application is accepted it requests to send a notification of the meeting and after the meeting it checks the eligibility and saves the record.

1.3 APPROVAL

Fig 1.3.1 Activity diagram for Approval

In this figure, we see that the provost makes a decision and the student is made resident accordingly.

1.3 APPROVAL



Fig 1.3.2 Sequence diagram for Approval

In this diagram we see that the provost gives his approval using the giveApproval() method and after that the control class gets the approval and makes the student resident using the makeResident() method from the class AddStudent.

1.3 APPROVAL



Fig 1.3.3 State diagram for Approval

In this diagram we see that the entity class Approver checks for approval and if it finds then makes the student resident.

1.4 CHECKING EMPTY ROOM

Fig 1.4.1 Activity diagram for checking empty room

In this figure, we see that the supervisor checks for an empty room. If he finds, then he prepares a list of empty rooms for the resident.

1.4 CHECKING EMPTY ROOM



Fig 1.4.2 Sequence diagram for checking empty room

In this diagram we see that the supervisor checks for an empty room using the checkroom() method and if any room is found, then the room no. is saved in the RoomAllocation class.

1.4 CHECKING EMPTY ROOM



Fig 1.4.3 State diagram for checking empty room

In this diagram we see that the entity class RoomChecker requests to the room database and if it finds it empty then it selects for requests select room.

1.5 SEAT ALLOCATION

Fig 1.5.1 Activity diagram for seat allocation

At first the supervisor logs into his account. Then he gets selected empty room form the control. And the seat is allocated for the student then.

1.5 SEAT ALLOCATION



Fig 1.5.2 Sequence diagram for seat allocation

In this diagram we see that the supervisor gets the no. of the empty room from the SeatAllocation control class and then a seat is allocated for the student using the allocateRoom() method.

1.5 SEAT ALLOCATION



Fig 1.5.3 State diagram for seat allocation

In this diagram we see that the entity class SeatAllocation requests for the allocated room.

1.6 INFORM STUDENT

Fig 1.6.1 Activity diagram for inform student

In this figure, we see that after login supervisor sets notification for student and sends notification. Student after login gets notification.

1.6 INFORM STUDENT



Fig 1.6.2 Sequence diagram for inform student

In this diagram we see that the supervisor sends a notification and when the student logs in to his account, he receives the notification.

1.6 INFORM STUDENT



Fig 1.6.3 State diagram for inform student

In this diagram we see that the entity class Inform checks if the room is allocated and if it is allocated then it requests for allocation notification.

1.7. INFORM AUTHORITY

Fig 1.7.1 Activity diagram for inform authority

In this figure, we see that the student first logs into his account .Then he sets notification and sends notification to the hall office. Supervisor logs into his account gets the notification about the students occupying seat and seat is allocated.

1.7. INFORM AUTHORITY



Fig 1.7.2 Sequence diagram for inform authority

In this diagram we see after occupying a seat, the student sets a notification using the setNotification() method. The control class gets the notification and sends the notification to the hall supervisor. The supervisor gets the notification using the getNotification() method from the Notification class.

1.7. INFORM AUTHORITY



Fig 1.7.3 State diagram for inform authority

In this diagram we see that the entity class InformAuthority checks if the room is occupied. If yes then it requests to inform the authority.

SUBSYSTEM2

MESS MANAGEMENT SYSTEM

SUB-SUBSYSTEM: MESS MANAGER ELECTION

There are four use cases

* Apply For Mess Management
* Check Whether Eligible
* Voting
* Approve and Announce

2.1.1. Apply For Mess Management

Fig 2.1.1.1 Activity diagram

In this diagram we see that the steps involved the students’ submitting an application for mess manager.

2.1.1. Apply For Mess Management



Fig 2.1.1.2 Sequence diagram

In this diagram we see that after getting logged in, a student has to follow to submit a application form for the mess manager. This process is finished when the application is received by the hall authority.

2.1.1. Apply For Mess Management



Fig 2.1.1.3 State diagram

In this diagram we see that the entity class ApplicationReceiver checks for any application available. If finds, it receives the application.

2.1.2 Check Whether Eligible

Fig 2.1.2.1 Activity diagram

At first provost login and get an application. After that decision is made and if he is approved his name is passed to the Candidate selector.

2.1.2 Check Whether Eligible



Fig 2.1.2.2 Sequence diagram.

In this diagram we see that the provost gets the application and then he made a decision which is approved and sends a notification to the student.

2.1.2 Check Whether Eligible



Fig 2.1.2.3 State diagram.

In this diagram we see that the entity class Decision checks for an application. If the application is accepted, it requests a meeting and sends a notification to the student and thereafter saves eligibility.

2.1.3 Voting

Fig 2.1.3.1 Activity diagram

After login student gen candidate list then he vote his favourite candidate,the vote is passed. After that vote is counted by counter.If he is elected provost is notified and the process is done.

2.1.3 Voting



Fig 2.1.3.2 Sequence diagram

In this diagram we see that students votes for the candidates and the votes is counted and the result is showed and saved.

2.1.3 Voting



Fig 2.1.3.3 State diagram

In this diagram we see that the class Counter checks for votes and starts counting.

2.1.4. Approve and Announce

Fig 2.1.4.1 Activity diagram

In this figure provost gets the result and gives approval and the result is announced thereby.

2.1.4. Approve and Announce



Fig 2.1.4.2 Sequence diagram

In this figure we see that the provost logs in and then get the result and approve the result. After that it sends the students the notification so that they can see the results.

2.1.4. Approve and Announce



Fig 2.1.4.3 State diagram

In this diagram we see that the class checks if the result is reviewed and approved and then it sends notification.

SUBSYTEM2

MESS MANAGEMENT SYSTEM

SUB-SUBSYSTEM: MESS MEAL SELECTION

There are three use cases

* Give menu list
* Select and submit within time
* Default meal if not submitted

2.1.1 GIVE MENU LIST

Fig 2.1.1.1 collaboration diagram

At first mess manager login then he gives a menu list. Then the form is filled up. Last selected menu is updated.

2.2.1 GIVE MENU LIST



Fig 2.1.1.2 Sequence diagram

In this diagram we see that the manager logs in to the account and fill up the forms and the control class gets the menu list and then update the list.

2.2.1 GIVE MENU LIST



Fig 2.1.1.3 State diagram

In this diagram we see that the entity class Menu checks for any menu available and if finds it selects the menu.

2.2.2 Select and Submit within Time

Fig 2.2.2.1 Activity diagram

At first students login then select menu.After selecting selected menu are passed to SelectedMenu.

2.2.2 Select and Submit within Time



Fig 2.2.2.2 Sequence diagram

In this diagram we see that the student logs in and then after getting the menu he select the menu and the menu is therefore selected.

2.2.2 Select and Submit within Time



Fig 2.2.2.3 State diagram

In this diagram we see that the entity class SelectedMenu checks for meal and if finds it submits menu list.

2.2.3 Default Meal if not Submitted

Fig 2.2.3.1 Activity diagram

At first menulist is checked.If someone don’t give any choice default menu is selected and default menulist is passed to default meal menu. 2.2.3 Default Meal if not Submitted



Fig 2.2.3.2 Sequence diagram

In this diagram we see that after certain time default menu is selected by the time actor.

2.2.3 Default Meal if not Submitted



Fig 2.2.3.3 State diagram

In this diagram we the entity class DefaultMenu requests for menu list and if any menu is not selected within time, it selects the default menu.