
Design Document

for

DigiCampus

Version 1.0

Prepared by

Group #: 1

Dishay Mehta	200341
Abhishek Pardhi	200026
Samarth Arora	200849
Ankur Kumar	200140
Shashwat Gupta	200923
Ananya Agrawal	200117
Girik Maskara	200387
Sarthak Kohli	200886
Aayush Kumar	200008
Aryan Vora	200204

Group Name: Achievers

dishaymehta952@gmail.com
abhipardhi326@gmail.com
samartharora03@gmail.com
ankurk04@gmail.com
guptashashwatme@gmail.com
ananyaagrawal704@gmail.com
girikmaskara552002@gmail.com
sarthak811kohli@gmail.com
aayushk0072@gmail.com
aryanvora23@gmail.com

Course: CS253

Mentor TA: *Aman Aryan*

Date: 13/02/22

CONTENTS	II
REVISIONS	II
1 CONTEXT DESIGN	1
1.1 CONTEXT MODEL	1
1.2 HUMAN INTERFACE DESIGN	1
2 ARCHITECTURE DESIGN	2
3 OBJECT-ORIENTED DESIGN	3
3.1 USE CASE DIAGRAM	3
3.2 CLASS DIAGRAM	3
3.3 SEQUENCE DIAGRAM	3
3.4 STATE DIAGRAM	3
4 PROJECT PLAN	4
5 OTHER REQUIREMENTS	5
APPENDIX A - GROUP LOG	6

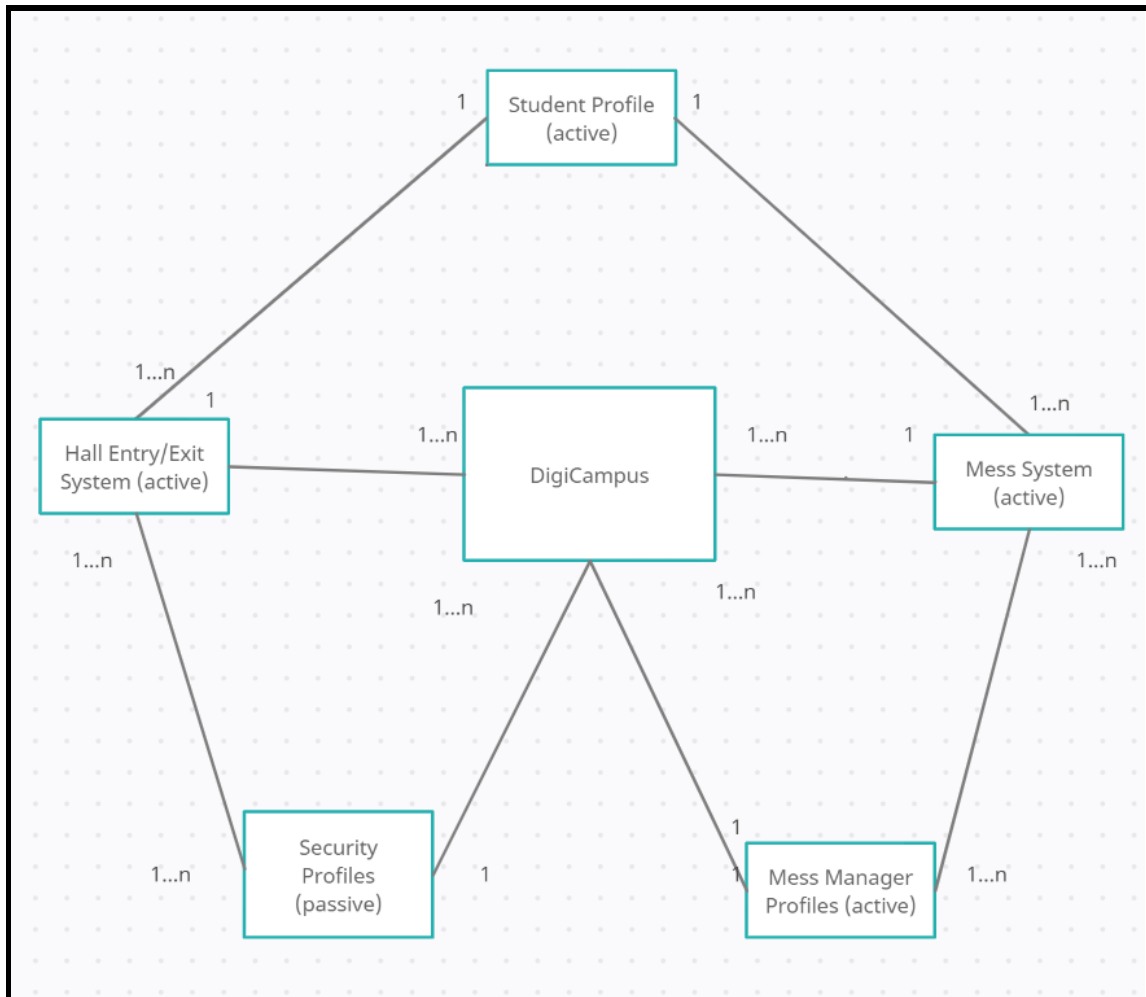
Revisions

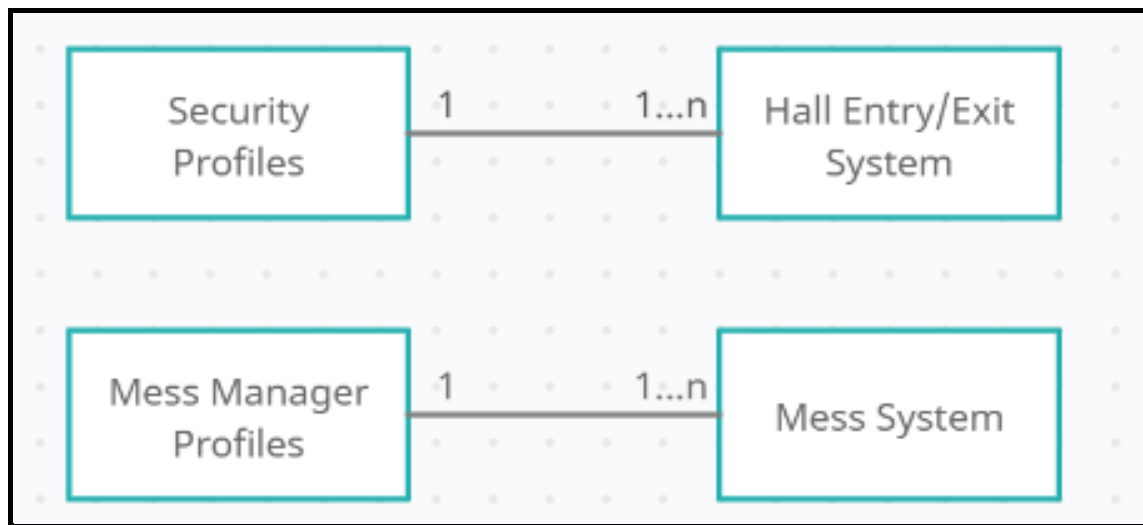
Version	Primary Author(s)	Description of Version	Date Completed
---------	-------------------	------------------------	----------------

1.0	Dishay Mehta,Abhishek Pardhi,Samarth Arora,Ankur Kumar,Shashwat Gupta,Ananya Agarwal,Girik Maskara,Sarthak Kohli,Aayush Kumar,Aryan Vora	Design Document created	15/02/22
-----	---	-------------------------	----------

1 Context Design

1.1 Context Model





1.2 Human Interface Design

The first version of the human interface looks like this:

- **Student Profile:**

The screenshot shows a web application interface for a student profile. At the top, there is a search bar and navigation links for 'Dashboard', 'Mess', and 'QR Code'. The main content area displays the student's profile information in a light blue theme. On the left is a placeholder for a profile picture. The profile details are organized into two columns:

Field	Value
Student Name:	Abhishek Pardhi
Student Roll no.	206969
Student Mobile No	+91 9876543210
Parent Hall	H-3
Email ID	apardhi20@iitk.ac.in
Parent Hall Room	A-123

At the bottom, there is a red button labeled 'Student ID' with a QR code icon, and a link above it that says 'Click to scan'.

- **Scanning Code At The Gate**



- **Entry/Exit Details**

The screenshot shows a web application interface with a light blue background. At the top, there is a navigation bar with a search bar labeled "Search for anything..." and a user profile icon. Below the navigation bar, there are three tabs: "Dashboard", "Menu", and "QR Code". The "QR Code" tab is currently selected. In the center of the page, there is a white rounded rectangle containing the following form elements:

- Entry in Hall 3** (Section Header)
- Enter Destination** (Text label)
- (Text input field) with a dropdown arrow icon on the right.
- Enter Roll number of resident** (Text label)
- (Text input field)
- ☐ (Checkbox) with the text "Check if you have a Laptop with you" next to it.

- Mess Menu and Extras

Search for anything...

Dashboard Mess QR Code

Menu

THURSDAY'S LUNCH

- Dal
1 serving, 100 g
- Rice
1 serving, 100 g
- Mix Veg
1 serving, 100 g

Add Extras

ITEMS PAYABLE

- Paneer Bhurji
1 serving, 150 g ₹30
- Anda Bhurji
1 serving, 100 g ₹30 ✓
- Rasgulla
1 serving, 100 g ₹30

Checkout

Main Lunch ₹0.00

EXTRAS

Anda Bhurji ₹30.00

CONFIRM

TOTAL PAYABLE ₹30.00

- Confirmation Page

Search for anything...

Dashboard Mess QR Code

Order Confirmed ✓

Order Summary

Main Lunch ₹0.00

EXTRAS

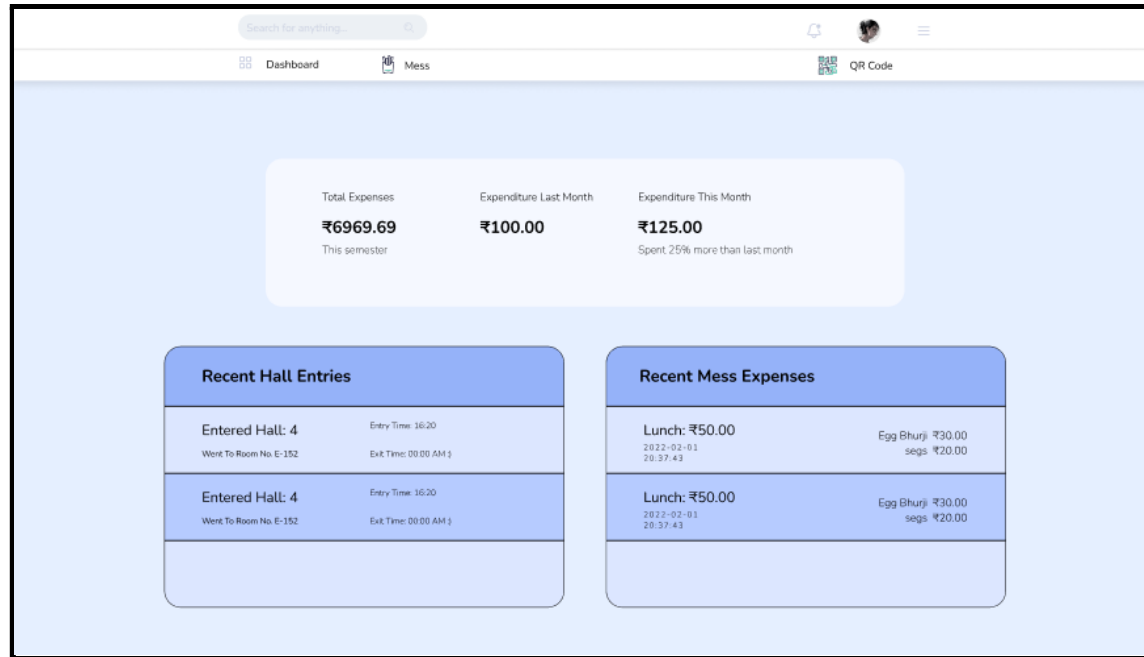
Anda Bhurji ₹30.00

Rasgulla ₹20.00

TOTAL PAYABLE ₹50.00

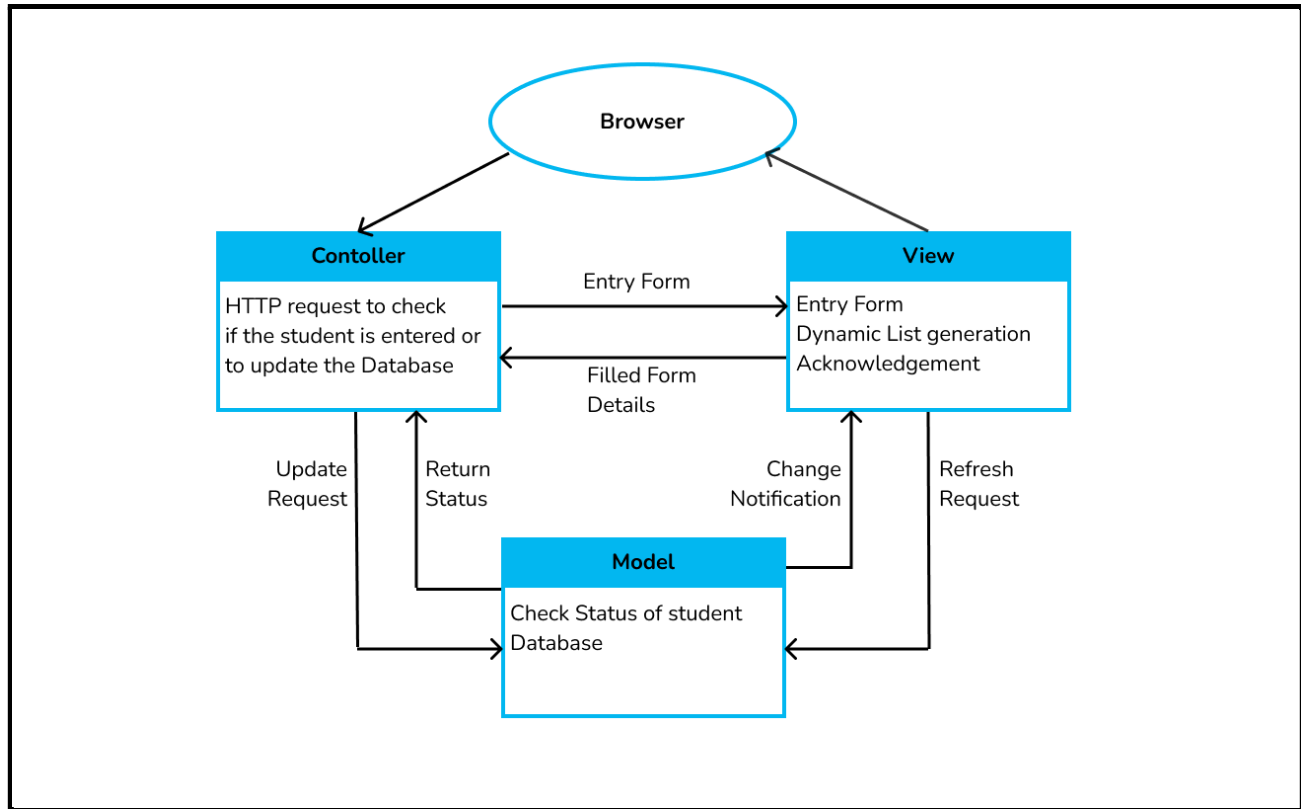
2022-02-01 20:37:43

- Student Dashboard



2 Architecture Design

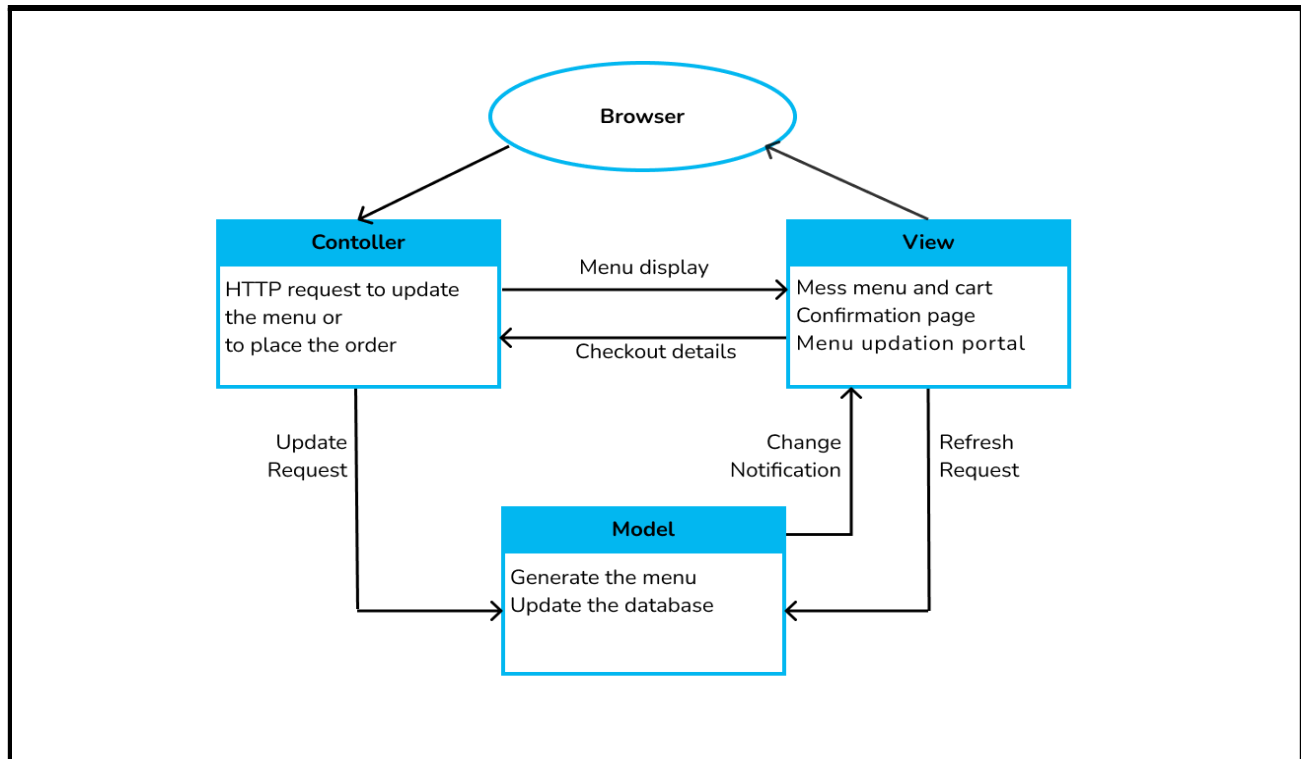
- For Entry/Exit :



The student scans the QR code → Control goes to the Controller which makes a request to check if the student is inside or outside → Request comes to the Model which checks for a student in the Database → If present, the student is given the exit acknowledgment and the dynamic database is manually updated / Else, Student is given entry form → Student fills entry form and details go to Controller which sends it to Model → Model updates the Student database marking student present inside.

The Dynamic List Generation makes it simpler for us to refresh the Student Database manually once the Entry/Exit of any student is done.

- For Mess :



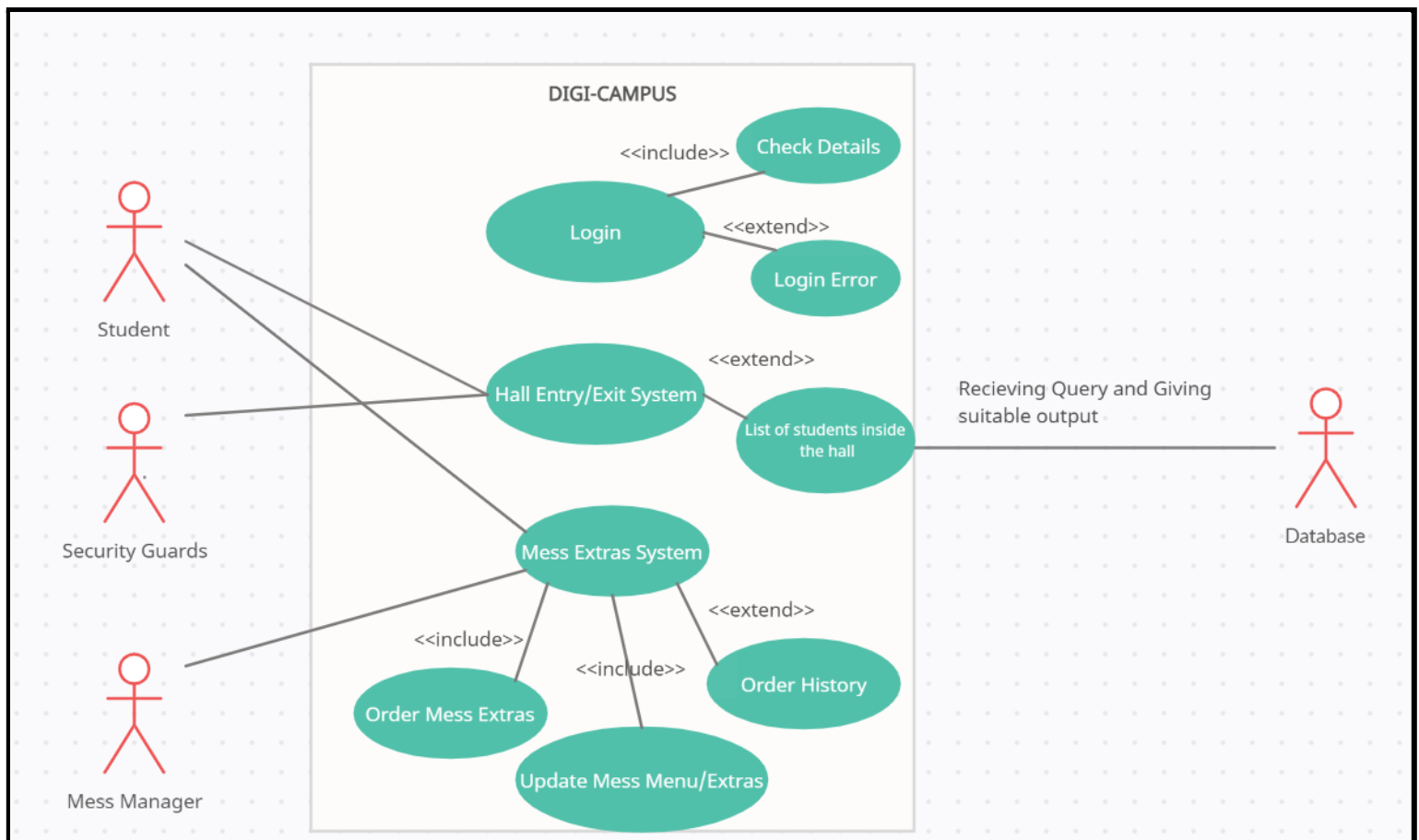
Here we have two functionalities :

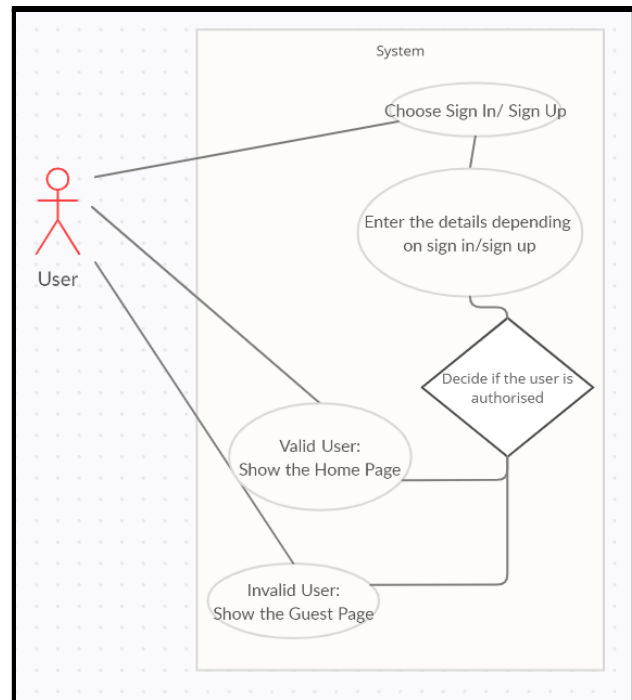
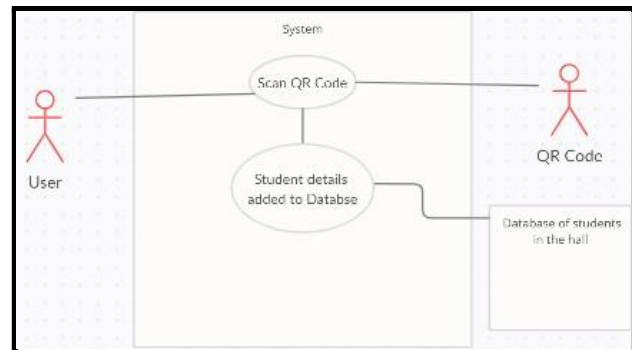
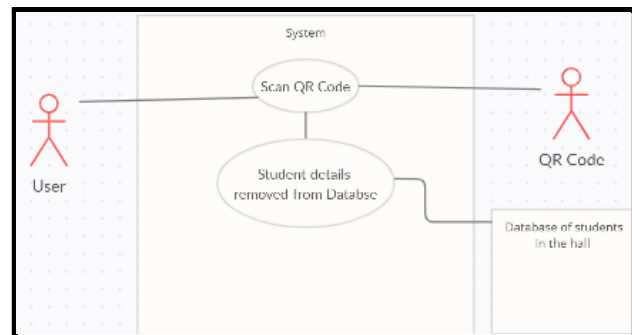
1. For Mess Manager: Mess manager sends an HTTP request to update the mess menu → Control goes to Model to Generate the menu → New menu is updated → Menu updation portal is displayed with an updated menu.
2. For Students: Student sends HTTP request to place an order → Control goes to view where mess menu and cart is displayed → Students place order in the cart and go to confirmation page → Change notification request is sent which updates student database → Checkout details is displayed to be shown at the mess.

3 Object Oriented Design

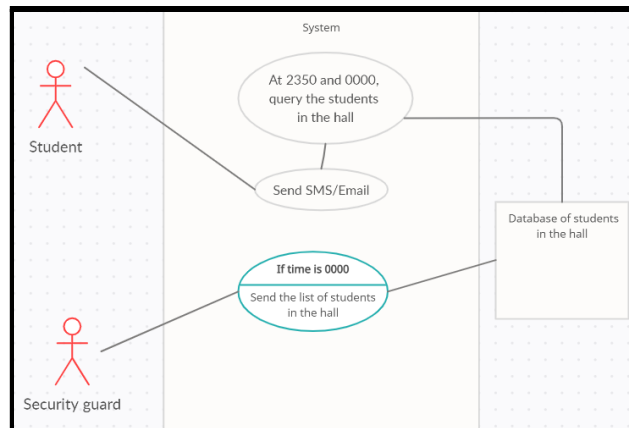
3.1 Use Case Diagrams

Use Case Diagram for the entire System:

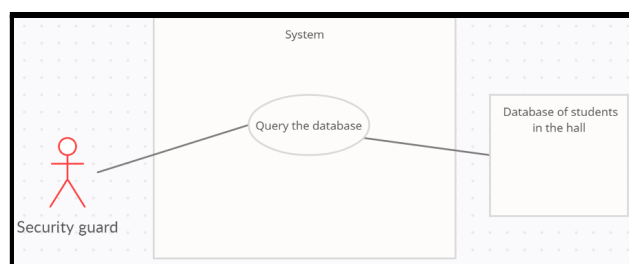


USE CASE DIAGRAMS FOR VARIOUS FUNCTIONALITIES:**USER CONFIGURATION****ENTERING A HALL****EXITING A HALL**

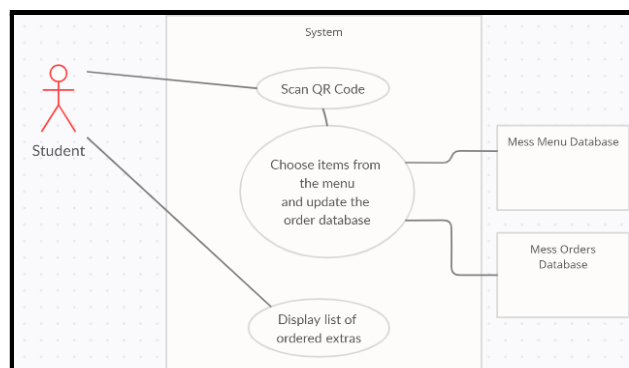
NOTIFICATION



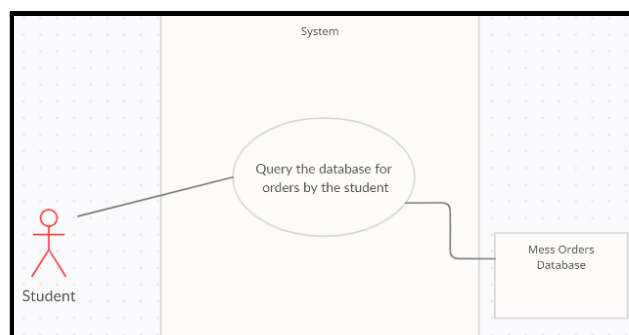
QUERY OF STUDENTS INSIDE THE HALL



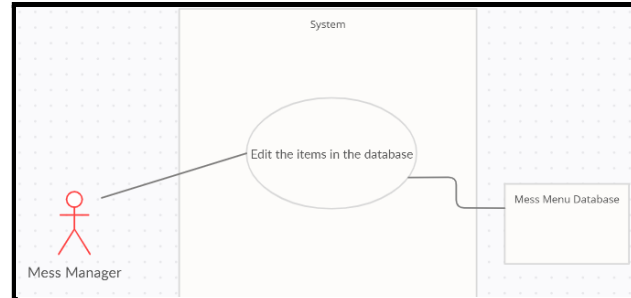
ORDERING MESS EXTRAS



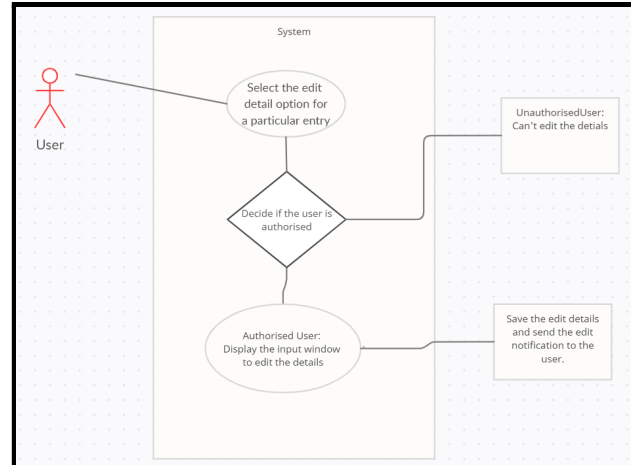
ORDER HISTORY



MENU UPDATION

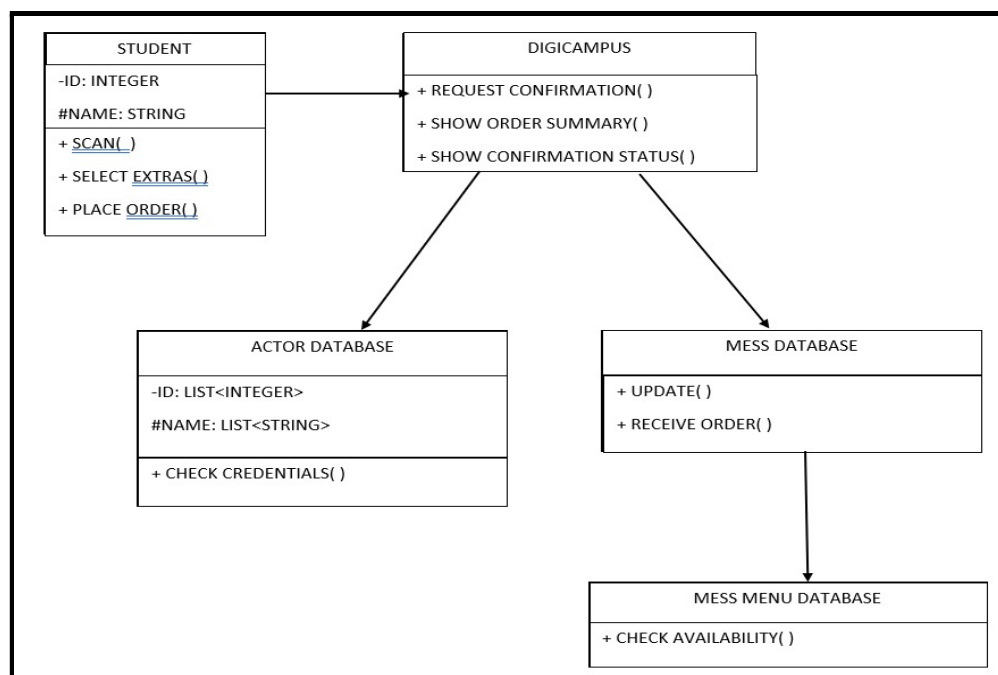


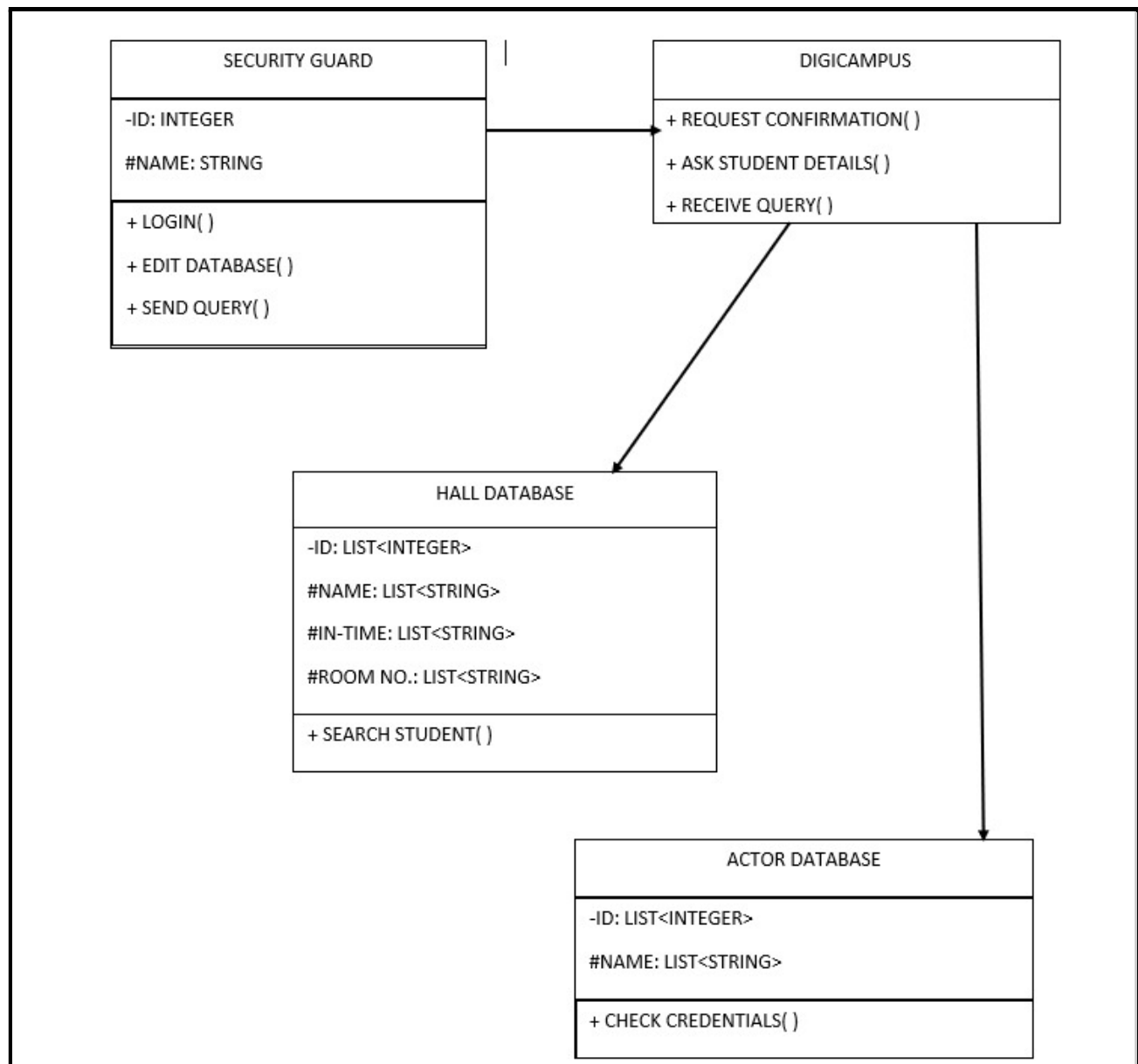
EDITING HALL ENTRY/EXIT DETAILS

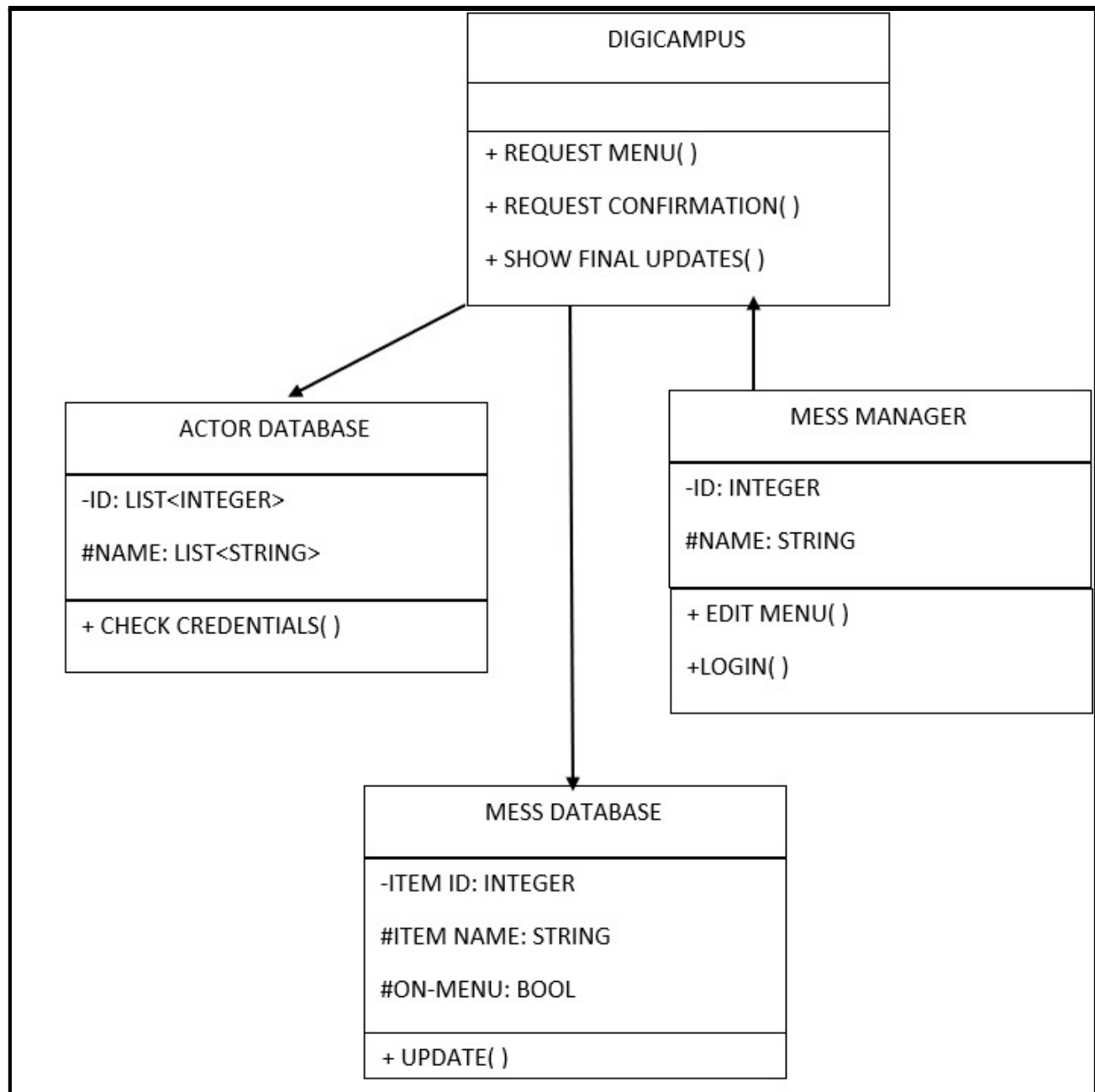


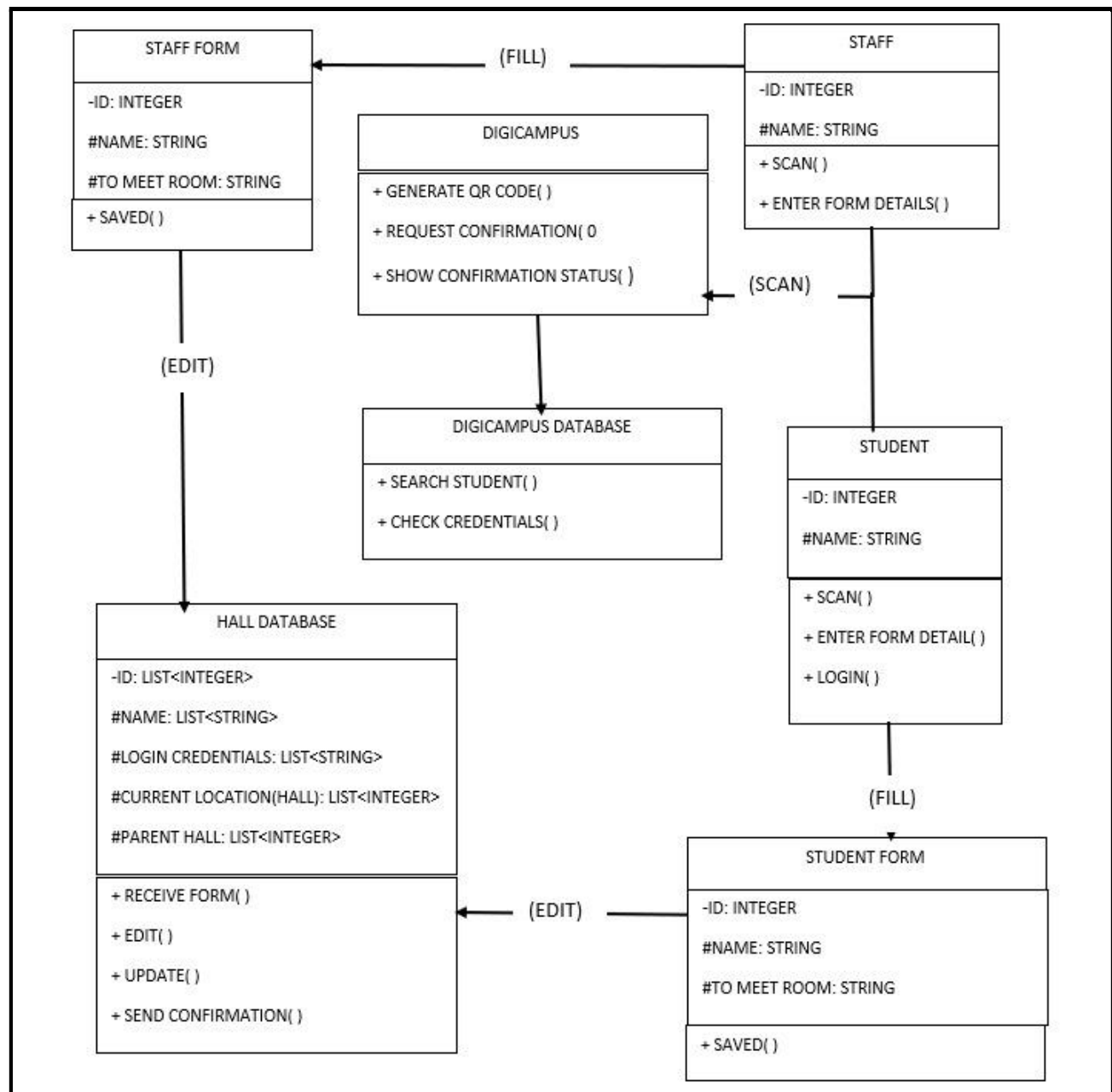
3.2 Class Diagrams

(A) FOR STUDENT:



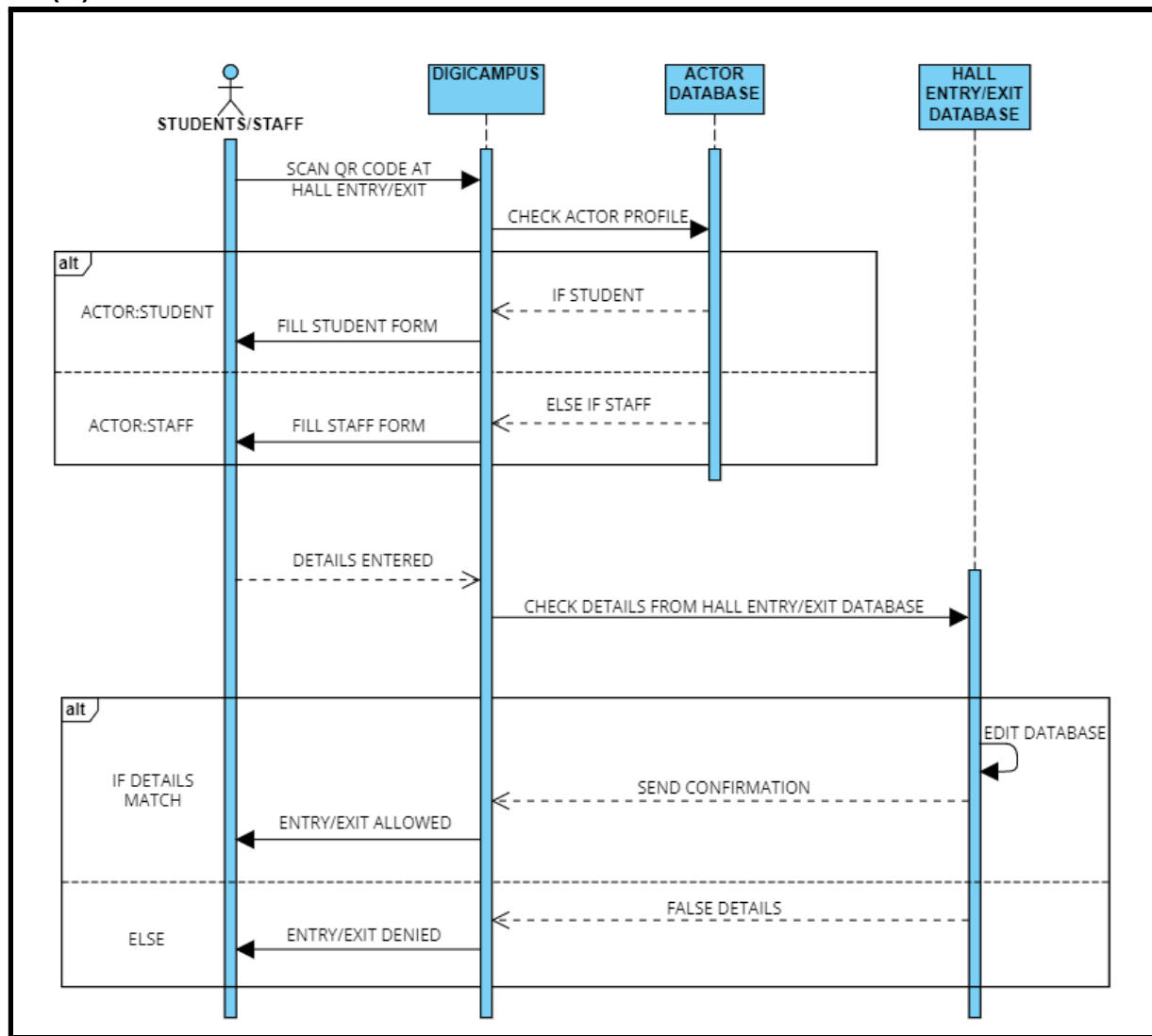
(B) FOR SECURITY GUARDS:

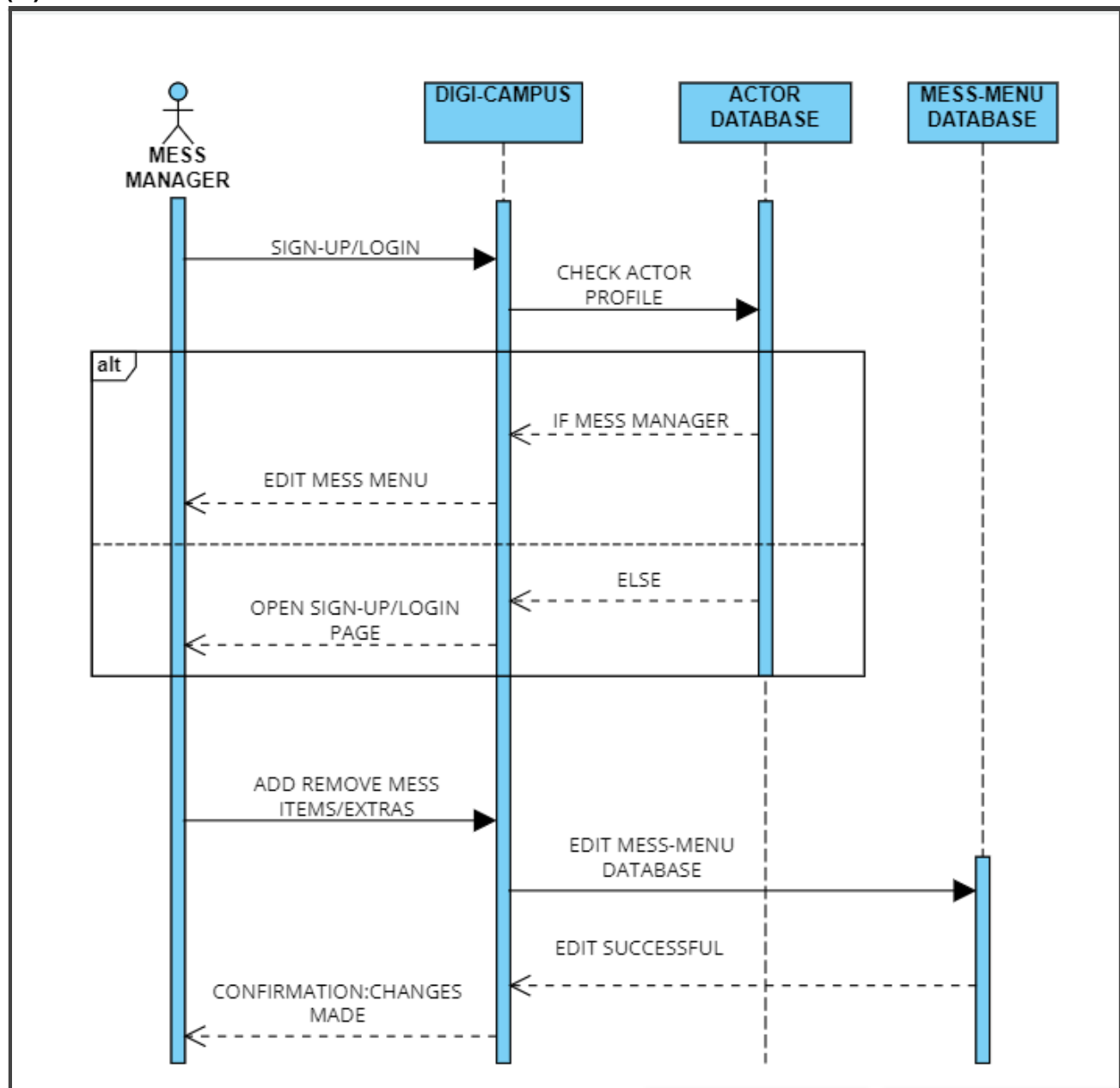
(C) MENU EXTRAS SYSTEM:

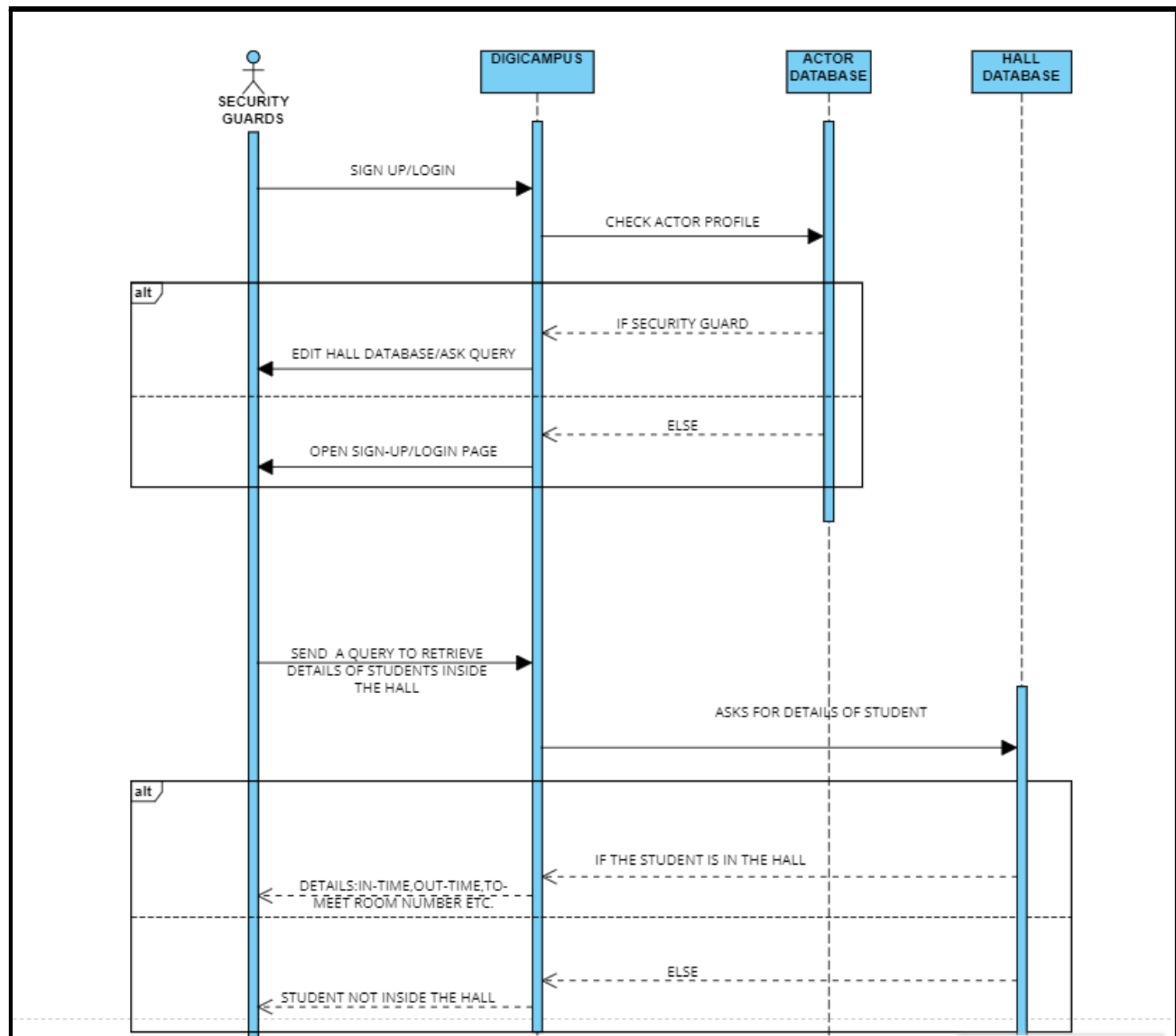
(D) HALL ENTRY SYSTEM:

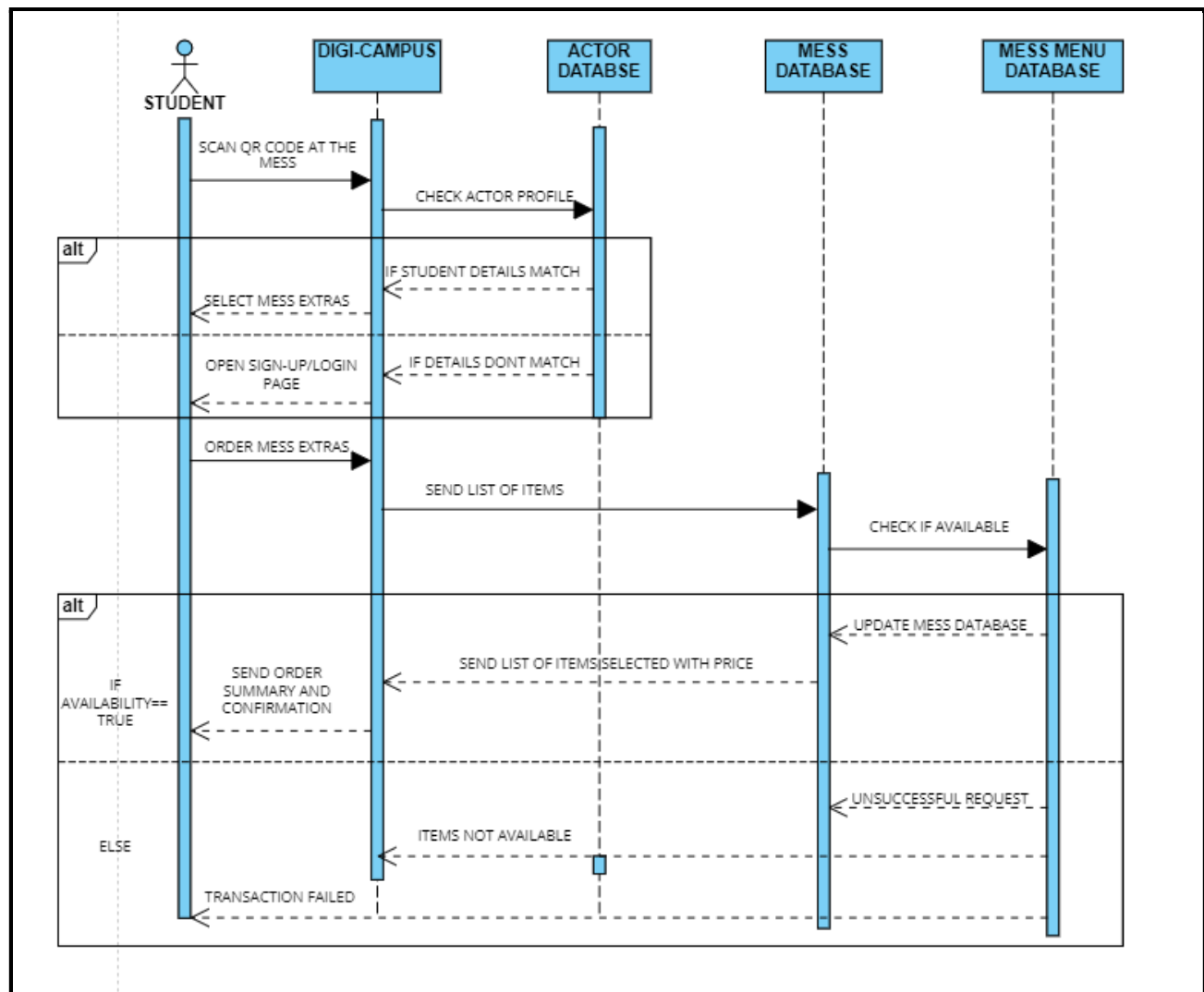
3.3 Sequence Diagrams

(A) ACTOR: STUDENT/STAFF-HALL/ENTRY-EXIT SYSTEM



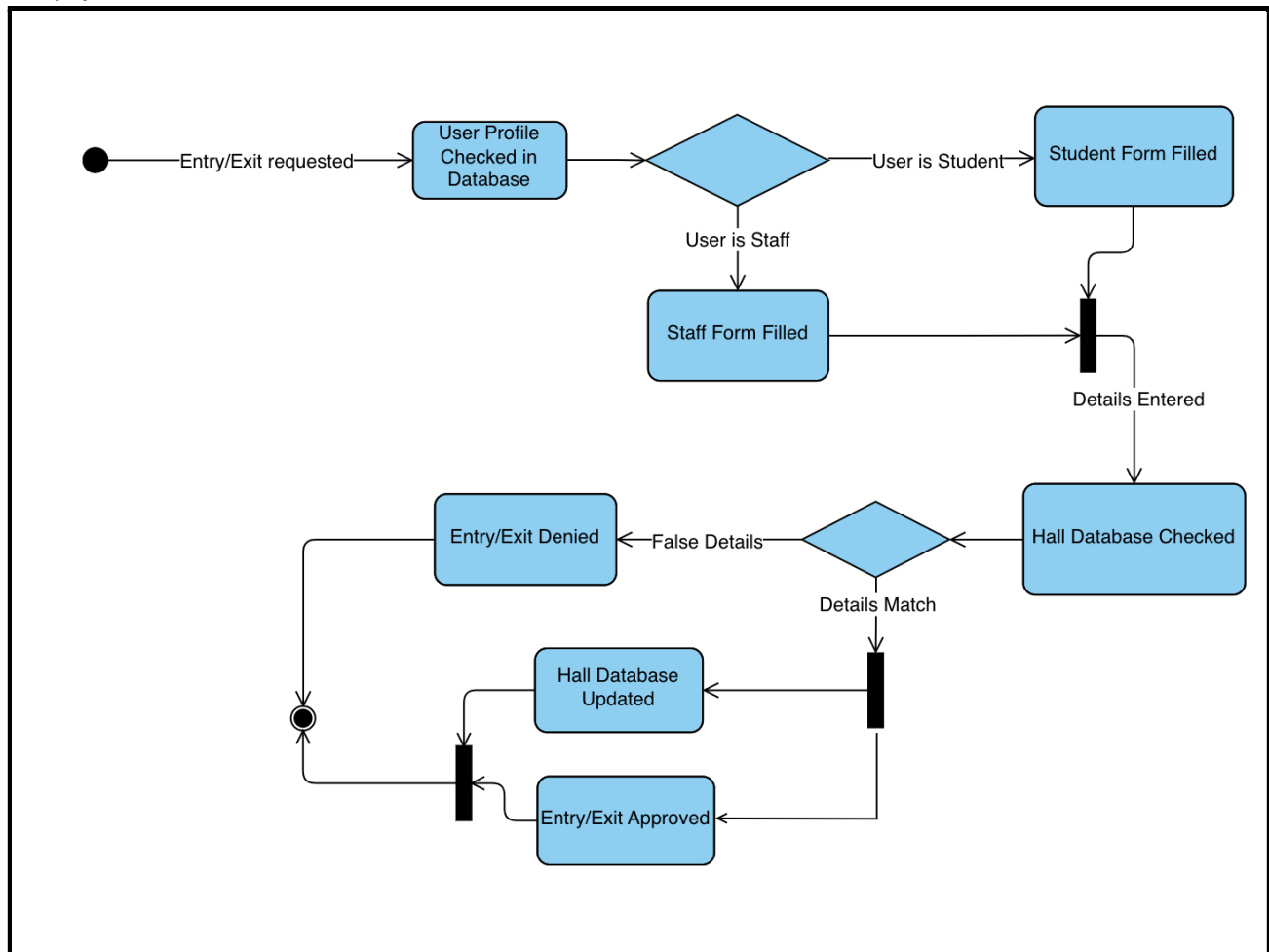
(B) ACTOR: MANAGER

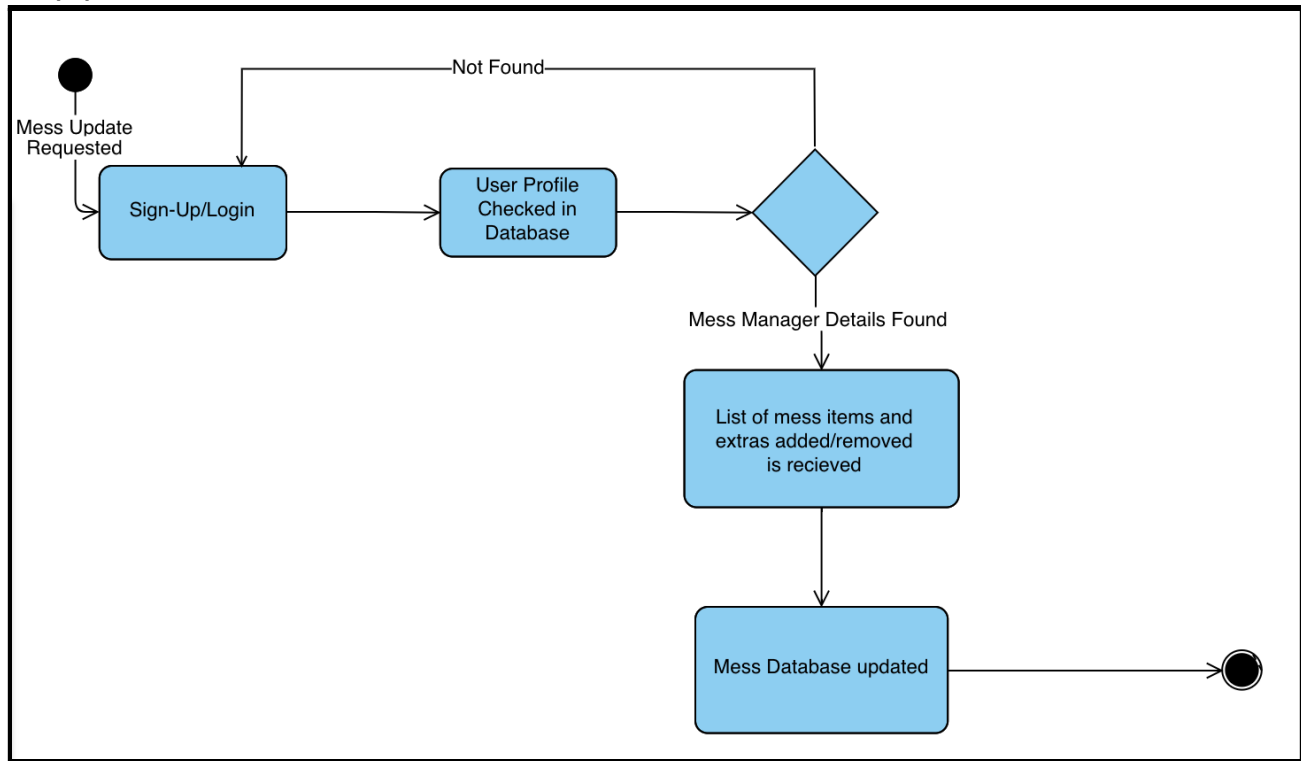
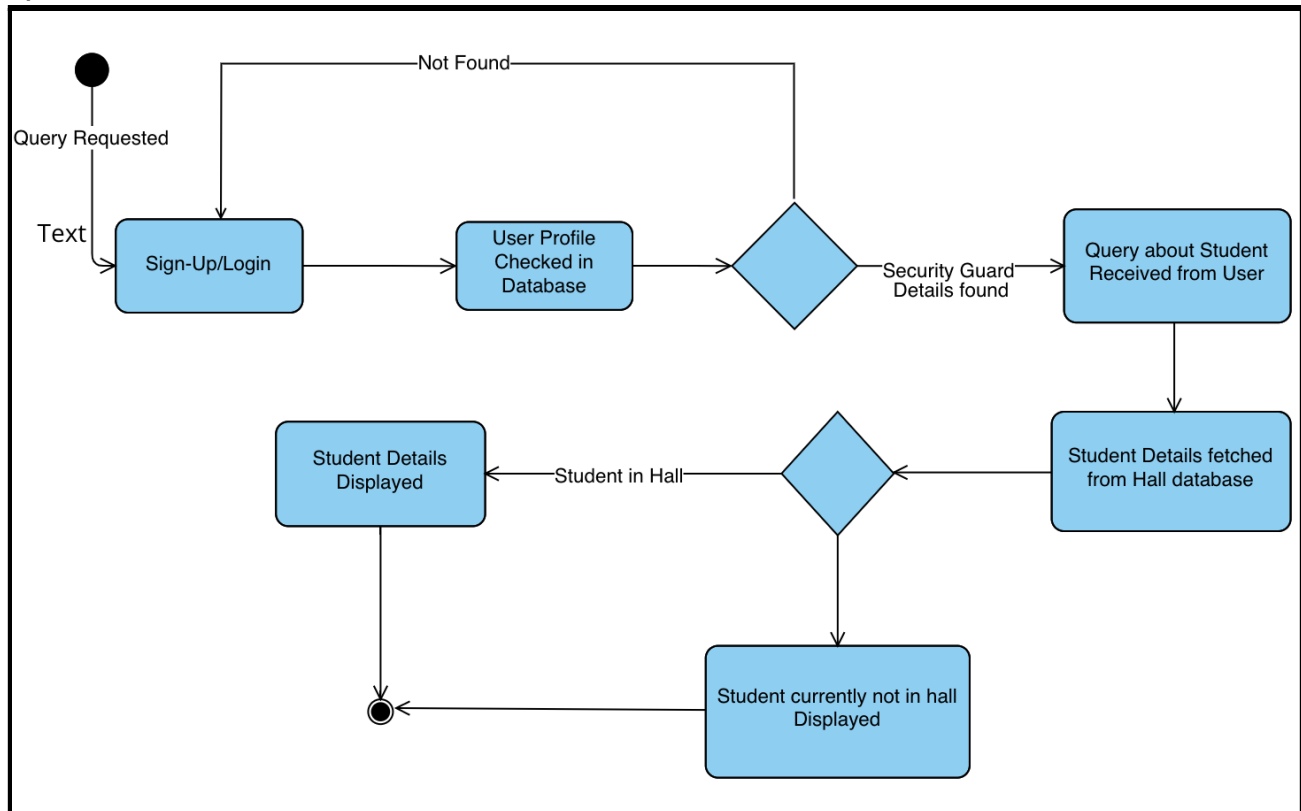
(C)ACTOR: SECURITY GUARDS

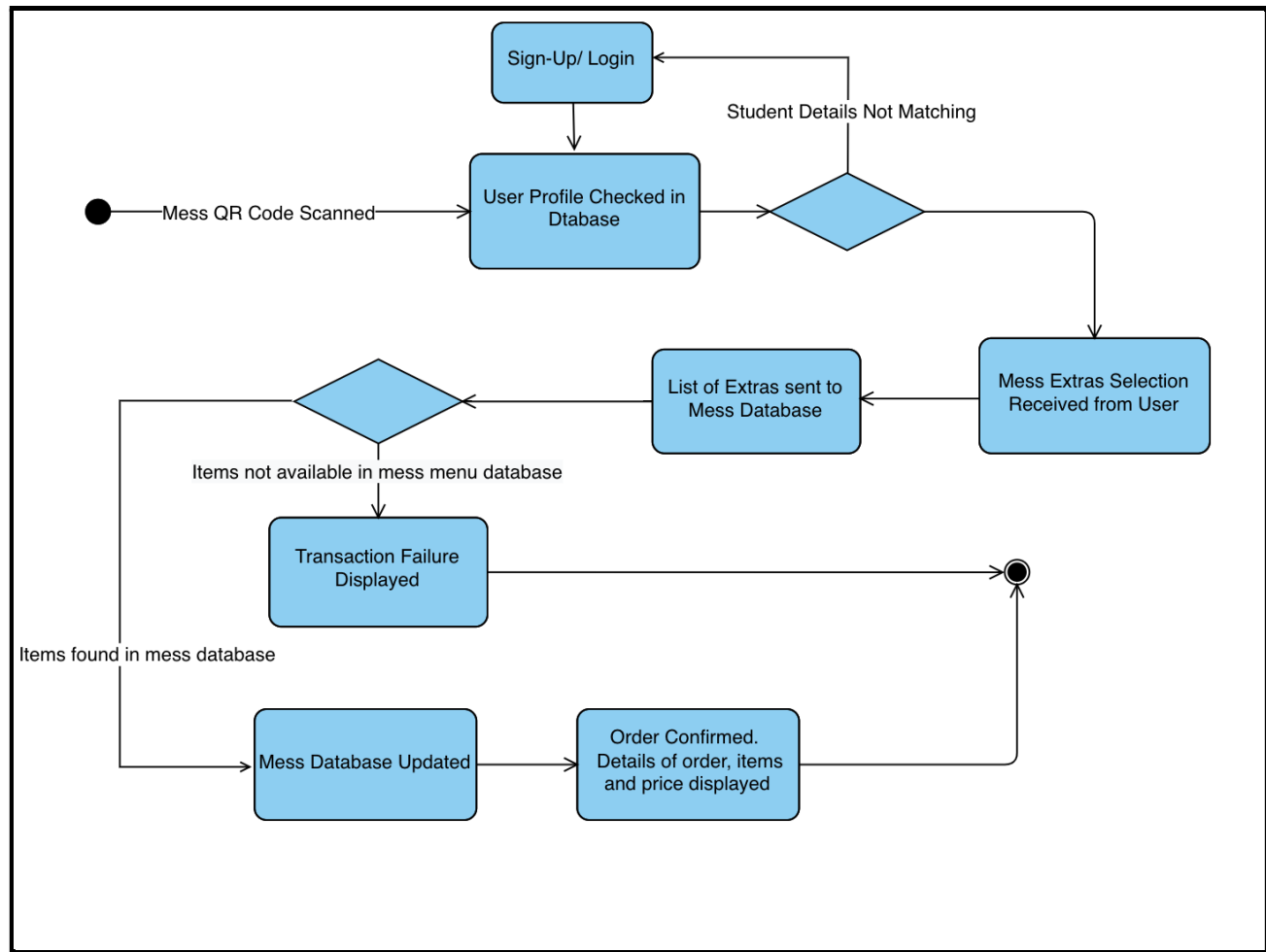
D)ACTOR: STUDENTS-MESS EXTRAS SYSTEM

3.4 State Diagrams

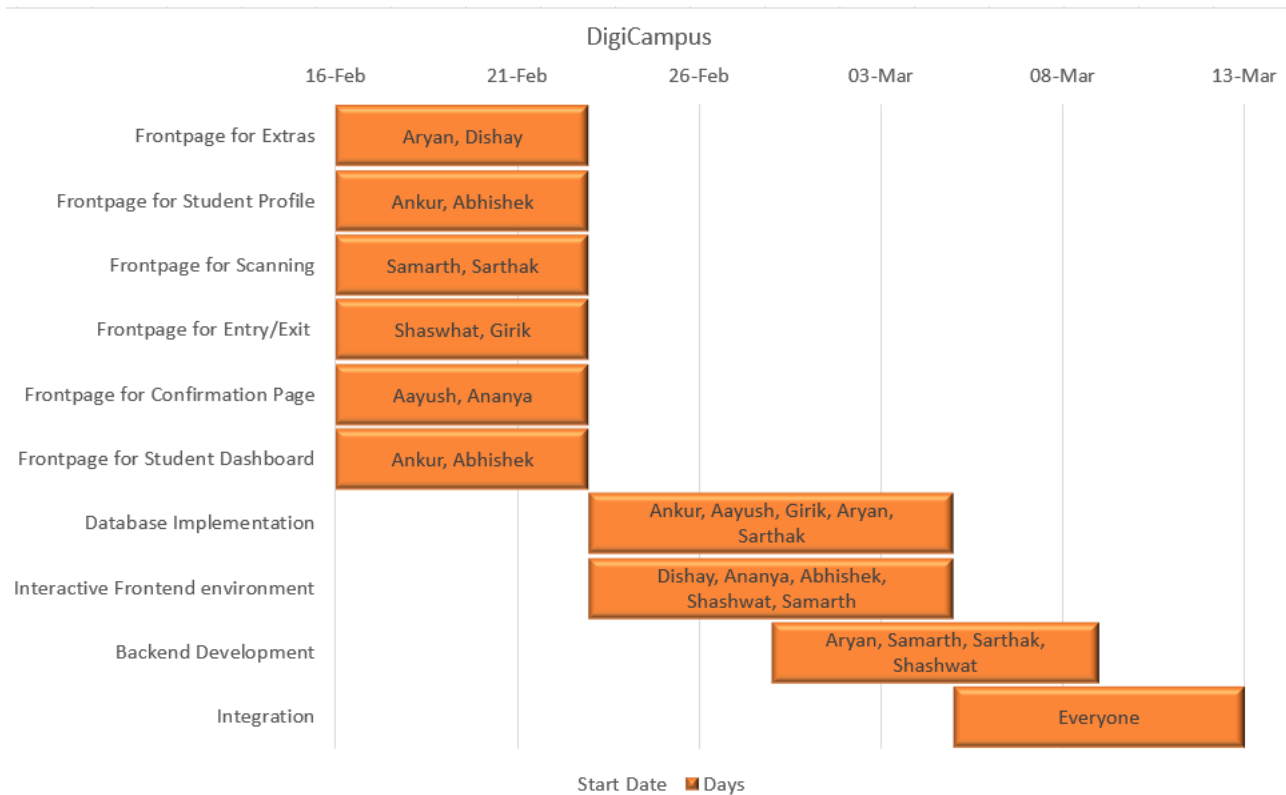
(A) ACTOR: STUDENT/STAFF-HALL/ENTRY-EXIT SYSTEM



(B) ACTOR: MANAGER**C) ACTOR: SECURITY GUARDS**

D)ACTOR: STUDENTS-MESS EXTRAS SYSTEM

4 Project Plan



TEST PLAN FOR OUR SOFTWARE:

- **ANALYZING THE PRODUCT:**

- ☐ Review the product documentation so as to be fully informed about the product details, target audience, and how the product will function on the ground level.
- ☐ If any point is unclear, we may interview customers or the target audience and make some changes in the software and documentation if they are needed at all.

- **TEST OBJECTIVES:**

- ☐ Finding as many software defects as possible.
- ☐ Creating bug-free software before release.

- **TEST STRATEGY:**

- ☐ **Defining the Scope of Testing:**

Defining clearly what things will be tested(hardware, software, integration) and metrics of testing.

- ☐ **Tentative priority of Testing Type:**

- (A) UNIT TESTING(will be done during the implementation phase as well)
- (B) INTEGRATION TESTING
- (C) SYSTEM TESTING
- (D) API TESTING
- (E) CODE IMPROVEMENT

☐ **Creating Test Logistics:**

- Preliminary tests will be done by our team (**Alpha testing**) in the priority as mentioned earlier.
- Preparation will be done for the beta testing of other groups' projects.
- **Beta testing** will be done by peer groups.
- The Testing will be done in the tentative time frame **14th March,2022-10th April 2022.**

☐ **Test Criteria:**

- Tentatively, the team has decided to choose Exit Criteria as Test Criteria. However, this is subject to change as per the team's requirements.

☐ **Test Deliverables:**

- The Team will submit a Beta Testing Report(of other group's project) and address to Beta testing Feedback.
- The team will finally submit Final Project Report.

5 Other Requirements

We will be using the IIT Kanpur servers and student databases for implementing the system.
The database can be imported from the Pingala servers.

Appendix A - Group Log

6th February: Aryan, Dishay, Sarthak, and Samarth meet to discuss the plan.

6th February: Groups divided to work on each section.

8th February: Aryan and Dishay work on Section 1.

9th February: Abhishek and Ankur completed Architecture Design.

10th February: Abhishek and Ankur completed Human Interface Design.

10th February: Aryan, Dishay, and Shashwat complete Section 1

12th February: Samarth and Sarthak finished Sequence Diagrams.

14th February: Aryan added Gantt Chart

15th February: Sarthak and Samarth made the test plan for the project.

15th February: Sarthak completed Use case diagrams.

15th February: Ananya completed Class diagrams.

15th February: Aayush completed state diagrams.