

| | |
|------------------------------|--|
| Course code and name: | F28SD – Introduction to Software Engineering |
| Type of assessment: | Individual |
| Coursework Title: | Designing A Software-Based System |
| Student Name: | Akshay Arunkumar Garg |
| Student ID Number: | H00338776 |

Declaration of authorship. By signing this form:

- I declare that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.
- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the [University's website](#), and that I am aware of the penalties that I will face should I not adhere to the University Regulations.
- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on [Academic Integrity and Plagiarism](#)

Student Signature: AkshayArunkumarGarg

Date: 07/03/2022

Designing a Software-Based System



Prepared by:
Akshay Arunkumar Garg

Prepared on:
7th March 2022

Person Identification:
H00338776

Programme:
BSc Computer Science (AI)

Course :
Introduction to Software Engineering (F28SD)

Campus:
Dubai

TABLE OF CONTENTS

| | |
|--|----|
| INTRODUCTION..... | 4 |
| TASK 1 – ASSUMPTIONS | 5 |
| TASK 2 – REQUIREMENTS | 5 |
| NON-FUNCTIONAL REQUIREMENTS | 5 |
| FUNCTIONAL REQUIREMENTS | 5 |
| TASK 3 – USE CASE | 6 |
| USE CASE DIAGRAM | 6 |
| USE CASE SPECIFICATIONS | 7 |
| TASK 4 - TRACEABILITY MATRIX..... | 11 |
| TASK 5 – CLASS DIAGRAM | 11 |
| TASK 6 – SEQUENCE DIAGRAMS | 12 |
| GUEST – MAIN FLOW DIAGRAM | 12 |
| GUEST – ALTERNATIVE FLOW DIAGRAM | 13 |
| STAFF – MAIN FLOW DIAGRAM | 14 |
| STAFF – ALTERNATIVE FLOW DIAGRAM | 15 |
| TASK 7 – ACTIVITY DIAGRAM | 16 |
| TASK 8 – STATE MACHINE DIAGRAMS | 17 |
| GUEST – STATE DIAGRAM..... | 17 |
| ESSENTIAL STAFF – STATE DIAGRAM | 17 |
| NON – ESSENTIAL - STATE DIAGRAM | 18 |
| TASK 9 – SCENARIO TEST CASES | 19 |
| TEST CASE DIAGRAM | 19 |
| ROUND TRIP SCENARIO TEST CASES..... | 20 |

INTRODUCTION

This report is about developing a software-based access control system which is known as Secure Occupancy System (SOS). This system is used to authorize the details of the individuals accessing the buildings in an institution. This system is connected to three external systems which are Staff Records System (SMS), Security Monitoring System (SMS), and Archive Access System (AAS).

There are nine tasks in total which I have designed specifications for the system. Starting with task one, I have made some assumptions for the system based on the specifications given to me on SOS and its external system. In task two, I have written some non-functional requirements and functional requirements that this system will take care of. There are approx. eight functional requirements which satisfy the specification of the system.

Keeping in mind, these functional requirements I have created a use case diagram that describes the requirements and I have also added the textual description of all the use cases which includes the main flow and the alternative flow of each case in task three. In task four, I have shown which use case fulfills the functional requirement.

In task five, I have created the UML diagram of the system which shows the classes and functions required for this system that will fulfill the functional requirements. They also include some fields required to store the data temporarily while validating the data coming from the keypads attached at the entrance of every building.

In task six, I have created a sequence diagram for guests and staff which shows how the system works in a flow. The diagram is available for both the main flow and the alternative flow. Following these sequence diagrams, I have created an activity diagram which is a flowchart that shows a flow of one activity to another activity which helps the user to access the building. This was a part of task seven in my report.

In task eight, I have created state machine diagrams, keeping three users in mind. The three users are guests, essential staff, and non-essential staff. There is one diagram for each user. And in task nine, I have drawn a test case diagram that shows all the possible conditions in the system when a user tries to access the building. And at last in task nine, I have shown some possible path flows of the system.

TASK 1 – ASSUMPTIONS

1. The staff for which guest is coming should be inside the building.
2. The guest should have a valid staff ID(SID) for registering in the system.
3. User details should be stored in the system before entering the building.
4. The user interface (UI) of the keypad is easy to understand and usable by the user or actor.
5. The guest cannot enter the building without the one-time code (OTC).
6. The staff entering the building knows the four-digit staff ID and a four-digit pin.
7. The access mode of the actor (who is accessing the building) should be asked first by the user interface of the keypad.
8. The new day is assumed to start at midnight.

TASK 2 – REQUIREMENTS

Non-Functional Requirements

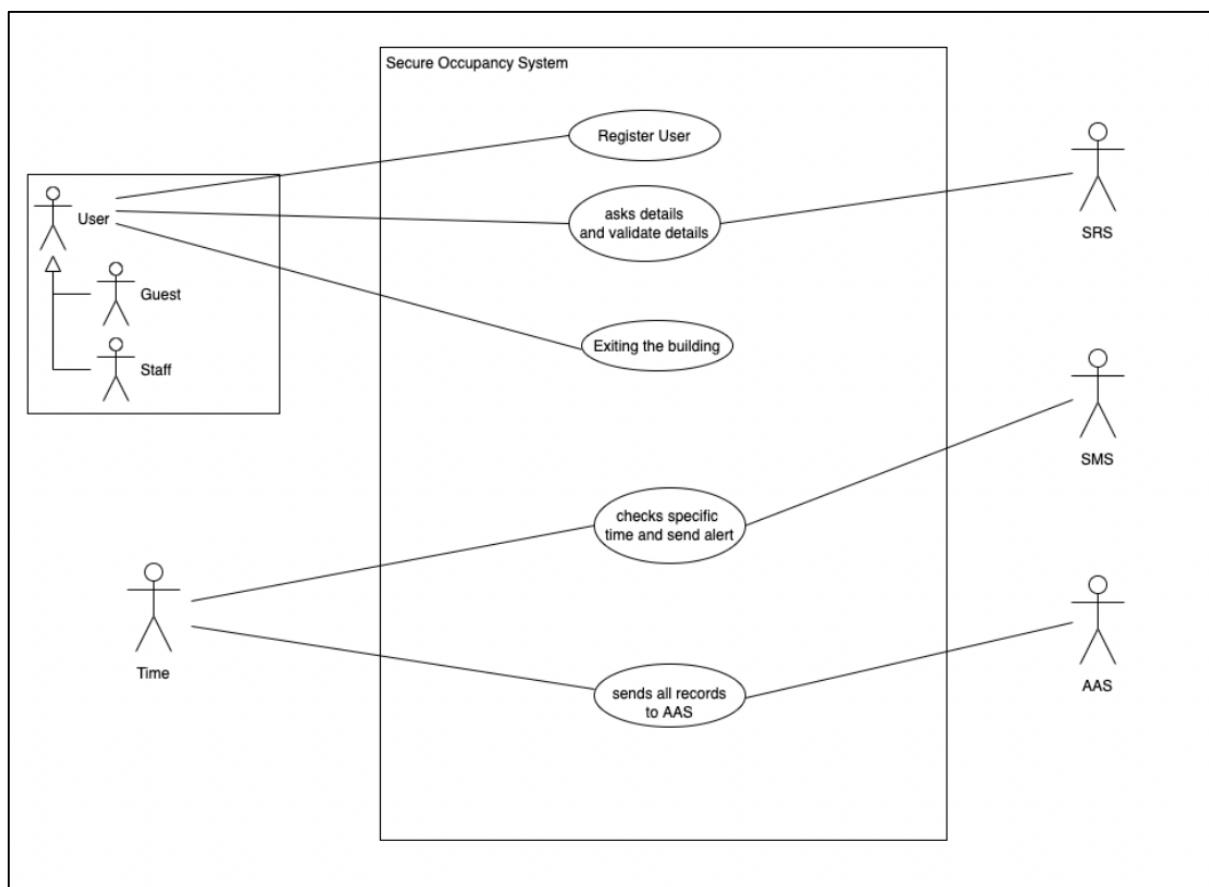
1. The user interface (UI) of the keypad is easy to understand and usable by the user or actor.
2. After validating the SID and pin of the staff and OTC of the user, immediate entry to the building is given.

Functional Requirements

| ID | Description | Priority |
|-----------|--|-----------------|
| FR1 | The details of the actor are stored in the system. If it is a guest then name, affiliation and SID should be stored. | M |
| FR2 | The system asks for the access mode from the actor and differentiates between the staff and the guests. | M |
| FR3 | The actors enter the details when asked by the system for accessing the building. | M |
| FR4 | The Secure Occupancy system (SOS) validates the SID and pin of the staff and OTC of the user with the Staff Records System (SRS). | M |
| FR5 | The system checks the working hours for the specific actor. | M |
| FR6 | If the user is inside the building after the core hours (9:00 – 22:00 for user) then an alert is generated, and SOS sends a message to Security Monitoring System (SMS). | M |
| FR7 | All the records of the day are archived by the SOS and sent to Archive Access System (AAS) when the new day starts at midnight. | M |
| FR8 | SOS stores the time at which actor leaves the building. | M |

TASK 3 – USE CASE

Use Case Diagram



Use Case Specifications

Use Case: Register User

ID: 1

Goal: Register a user with the Secure Occupancy System (SOS)

Primary actor: user

Secondary actor(s): None

Preconditions:

1. SOS is functioning correctly;
2. The user should have the Staff id to whom they are visiting;

Postconditions:

1. The user is registered with the SOS
2. The user is issued with the One Time Code (OTC)

Main flow:

1. *SOS requests user's details like name, affiliation, and other*
2. *User provides all the details and staff ID for whom they are visiting*
3. *User is registered in the system*
4. *User requests SOS for the OTC*
5. *SOS provides one time code to user which they can use to access the building*

Alternative flows:

2a. *Provided user details are invalid*

1. *SOS reports an error*
2. *Return to step 1 of main flow*

2b. *Provided staff id is invalid*

1. *SOS reports an error that provided staff id is invalid*
2. *Return to step 1 of main flow*

Use Case: asks details and validate details

ID: 2

Goal: User enters the details to access the building and validate the details

Primary actor: User

Secondary actor(s): Staff Records System (SRS)

Preconditions:

1. Guest has One Time Code (OTC);
2. Staff has the four-digit staff Id and PIN;

Postconditions:

1. The OTC and SID has been validated
2. Guest and the staff have access to the building

Main flow:

1. *User tries to access the building*
2. *Keypad requests user to select the access mode from two options guest and staff*
3. *User selects the access mode on the keypad*
4. *If the user selects access mode as guest*
 - 4.1. *Then system asks for eight-digit one-time code (OTC)*
 - 4.2. *SOS sends OTC and SRS validates the code and if the OTC is valid then it returns the staff id of the host*
5. *If the user selects access mode as staff*
 - 5.1. *Then system asks for the four-digit staff id and pin*
 - 5.2. *Then SOS sends the staff id and pin to SRS, and it validates the details and returns whether they are correct or not*
6. *SOS records the time at which user is entering the building*
7. *The access is granted to the user*

Alternative flows:

4.2.a. *The details entered by the guest is wrong*

1. *SOS reports an error*
2. *SOS requests to enter details again*
3. *Return to step 4 of main flow*

5.2.a. *The details entered by the staff is wrong*

1. *SOS reports an error*
2. *SOS requests to enter details again*
3. *Return to step 5 of main flow*

Use Case: Exiting the building

ID: 3

Goal: To store the time at which user leaves the building.

Primary actor: User

Secondary actor(s): none

Preconditions:

1. User is already inside the building;

Postconditions:

1. User is no more inside the building

Main flow:

1. *User requests to leave the building*
2. *SOS records the time at which they are leaving*
3. *User leaves the building*

Alternative flows:

none

Use Case: Checks specific time and sends alert

ID: 4

Goal: To check whether the user inside the building have access to building at that time.

Primary actor: Time

Secondary actor(s): Security Monitoring System (SMS)

Preconditions:

1. User is inside the building;

Postconditions:

1. *Alert is sent to SMS*

Main flow:

1. *It takes the current tim.*
2. *If the current time is before 9:00 and after 22:00, and if the user is guest, then SOS sends alert to SMS*
3. *If the current time is before 8:00 and after 22:00, and if the staff is non-essential staff, then SOS sends alert to SMS*

Alternative flows:

none

Use Case: sends all records to AAS

ID: 5

Goal: To send all the records of the day to AAS and deleted from SOS

Primary actor: Time

Secondary actor(s): Archive Access System (AAS)

Preconditions:

1. All the records of the day are stored by SOS;

Postconditions:

1. All the records are deleted from the system of SOS

Main flow:

1. *At 23:59, SOS sends all the records of individuals entering/exiting the building to AAS*
2. *AAS confirms that all the records have received to them*
3. *After confirmation from AAS, SOS deletes all the records from their system.*

Alternative flows:

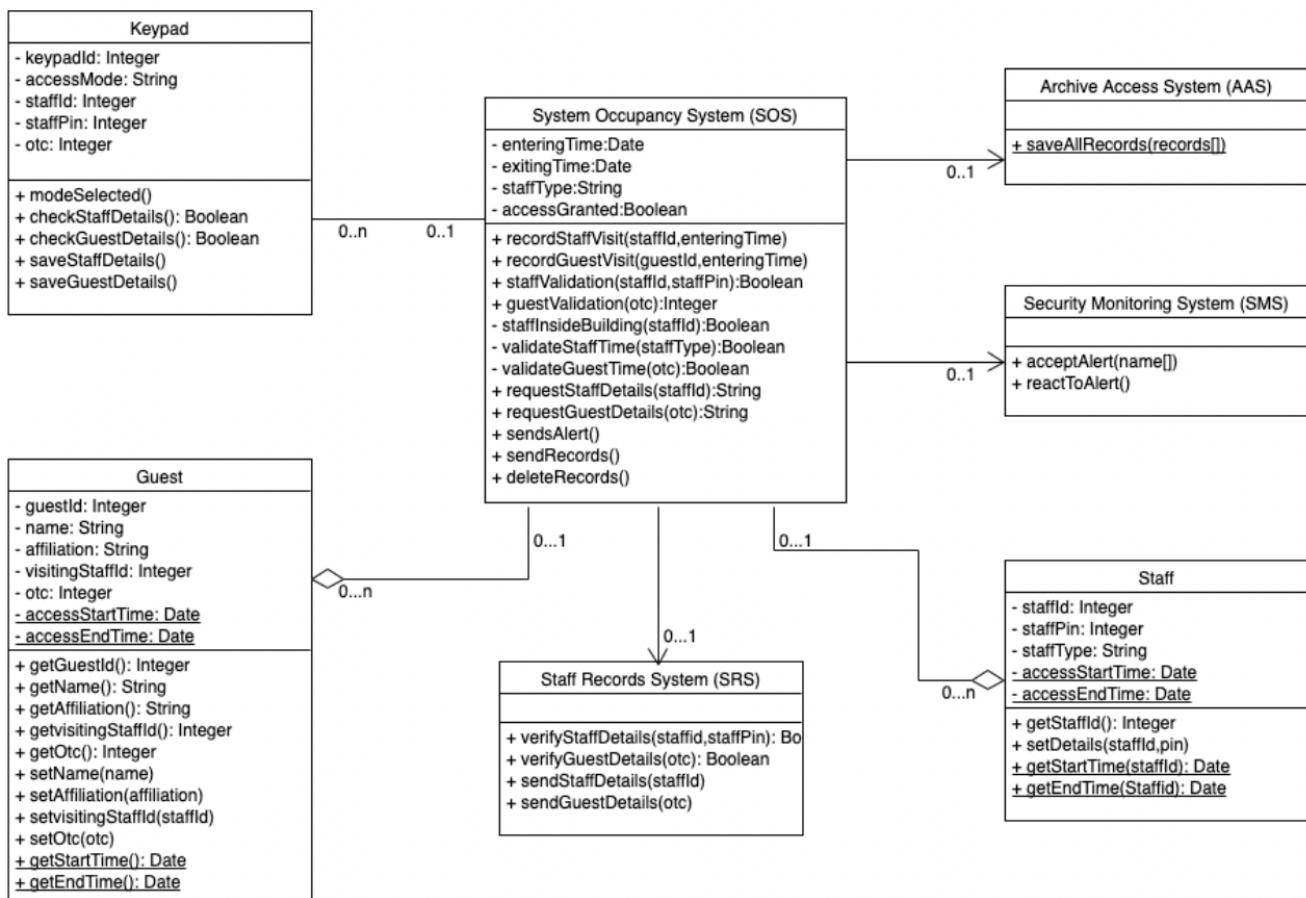
2a. AAS doesn't receive all the records

1. *AAS sends message that all the records are not received*
2. *Return to step 1 of main flow*

TASK 4 - TRACEABILITY MATRIX

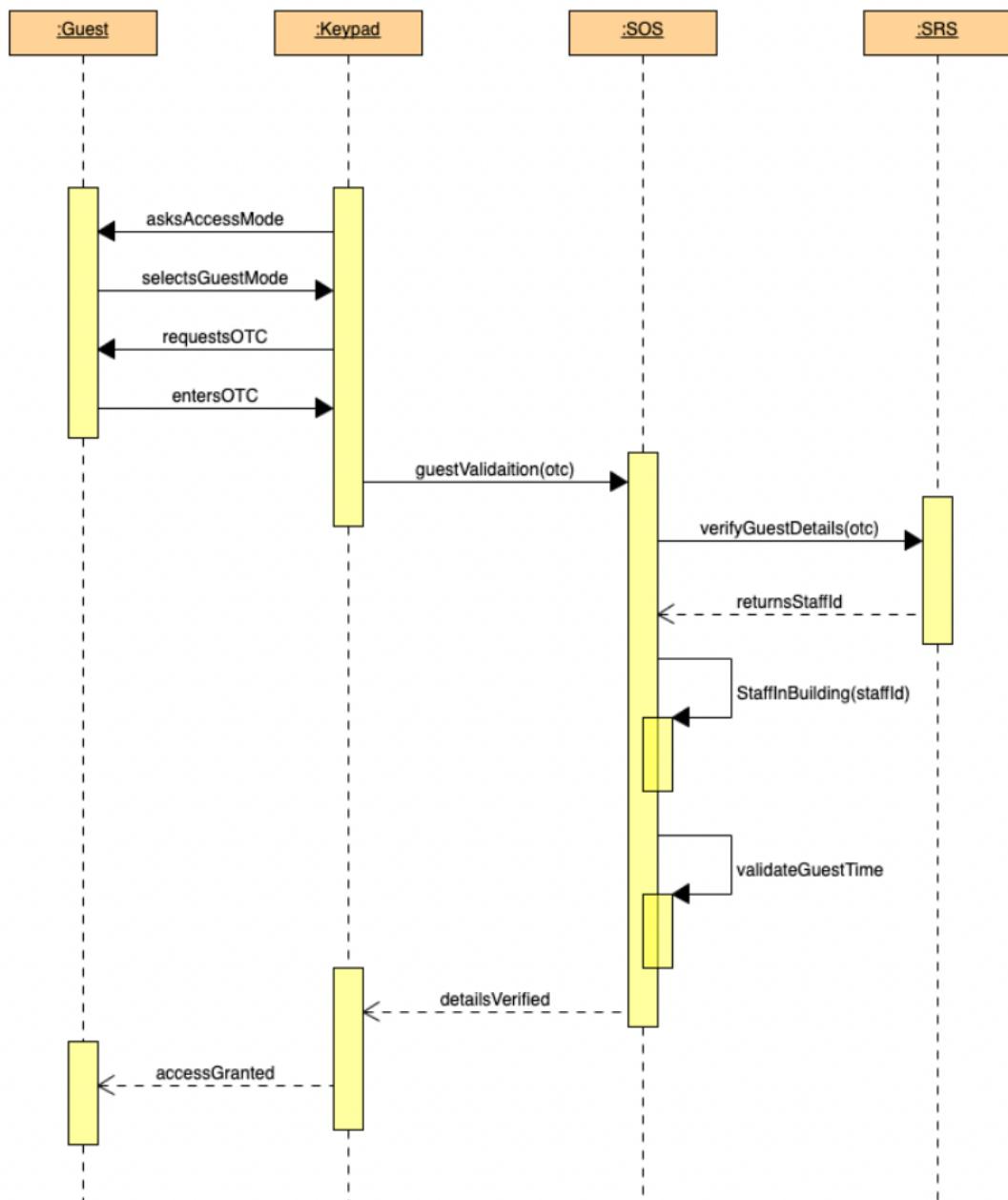
| FR/UC | UC1 | UC2 | UC3 | UC4 | UC5 |
|-------|-----|-----|-----|-----|-----|
| FR1 | x | | | | |
| FR2 | | x | | | |
| FR3 | | x | | | |
| FR4 | | x | | | |
| FR5 | | | | x | |
| FR6 | | | | x | |
| FR7 | | | | | x |
| FR8 | | | x | | |

TASK 5 – CLASS DIAGRAM

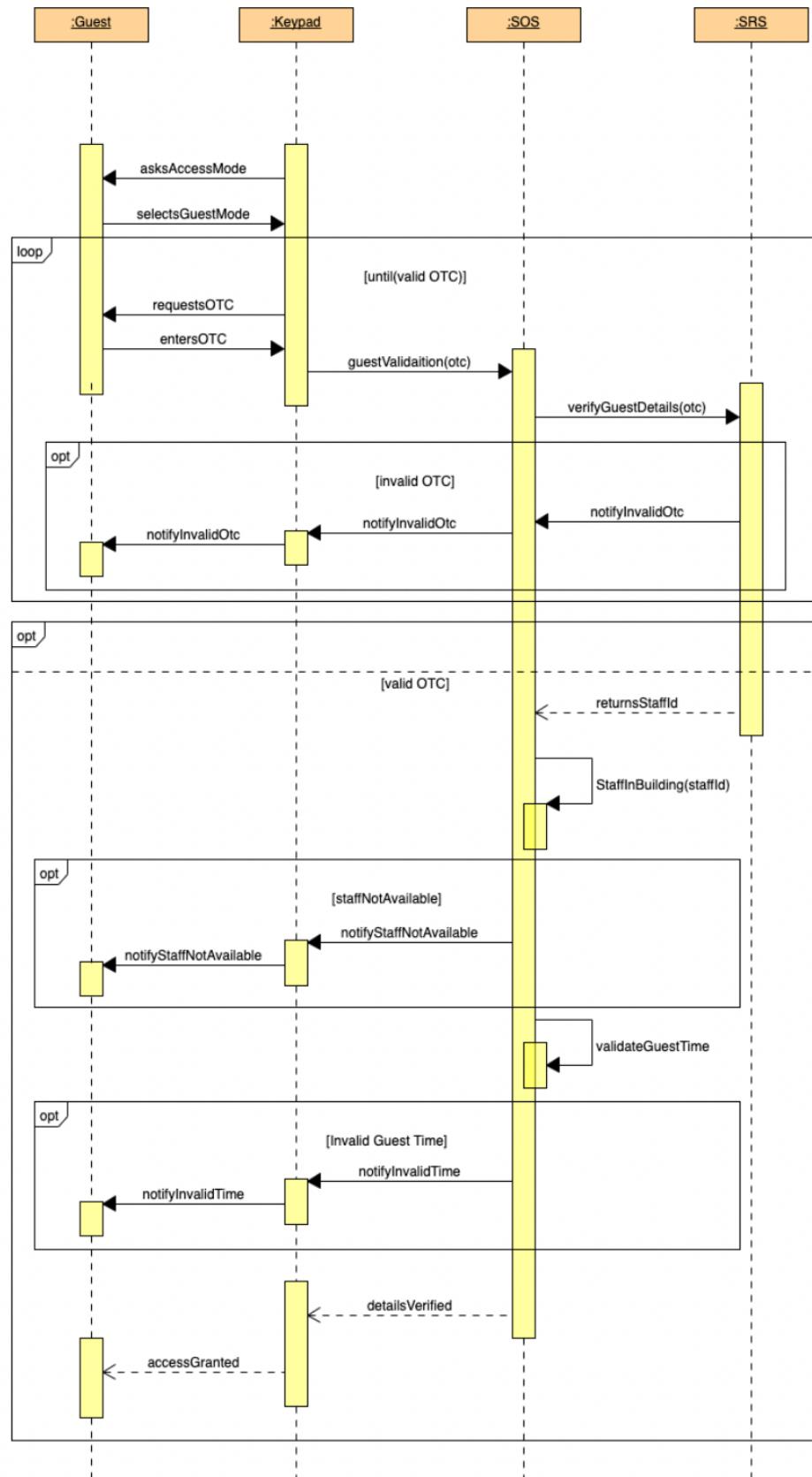


TASK 6 – SEQUENCE DIAGRAMS

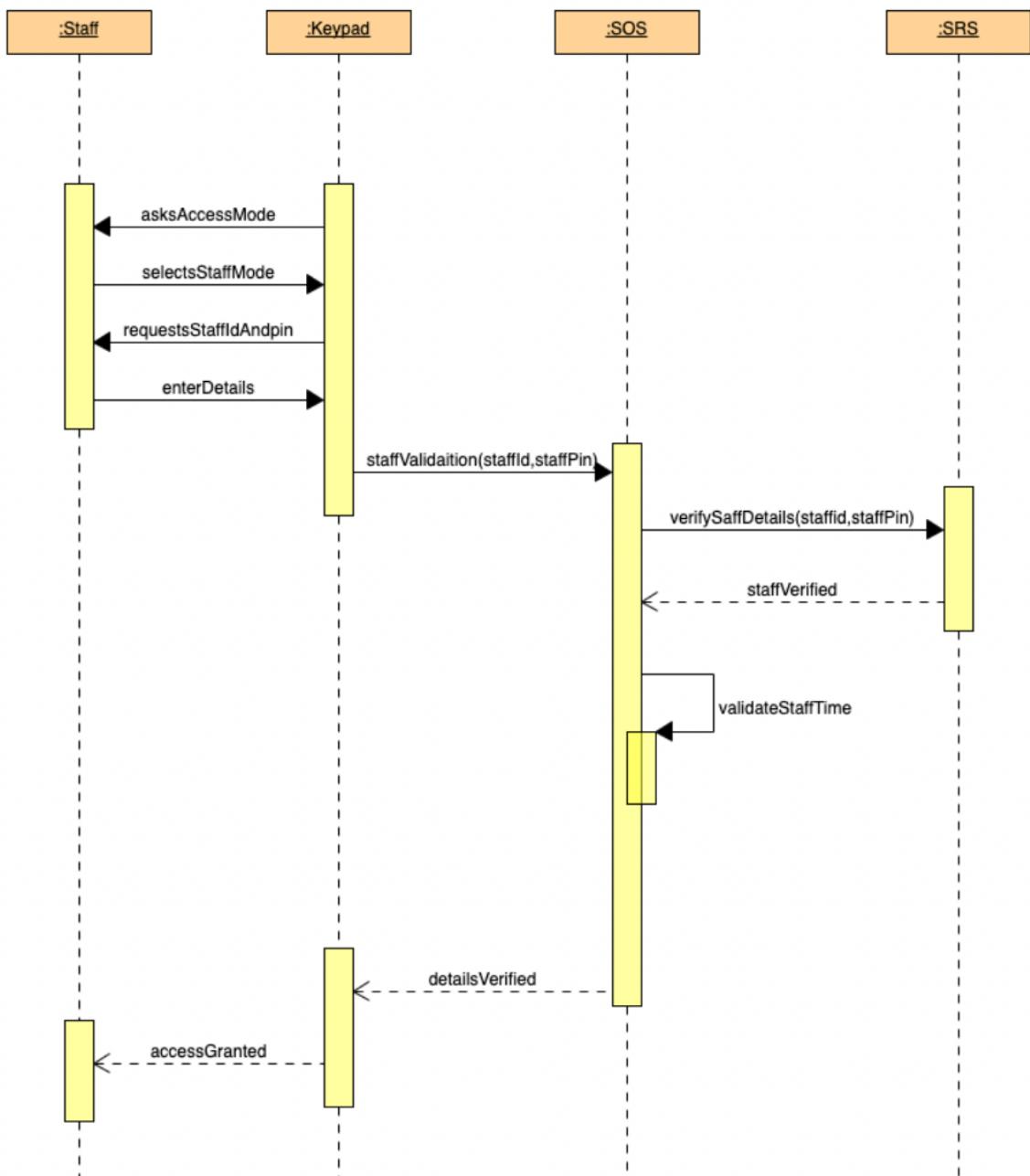
Guest – Main Flow Diagram



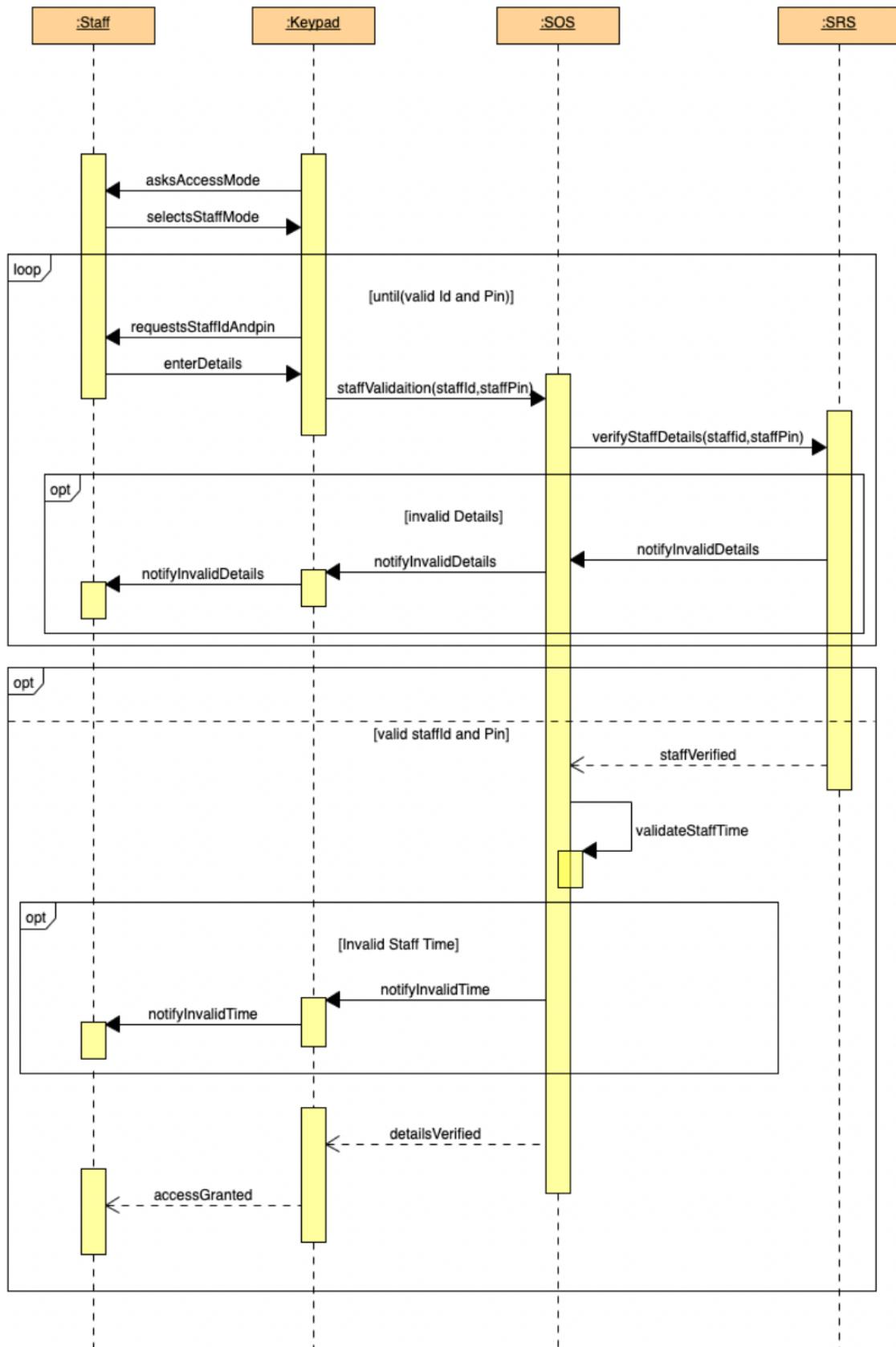
Guest – Alternative Flow Diagram



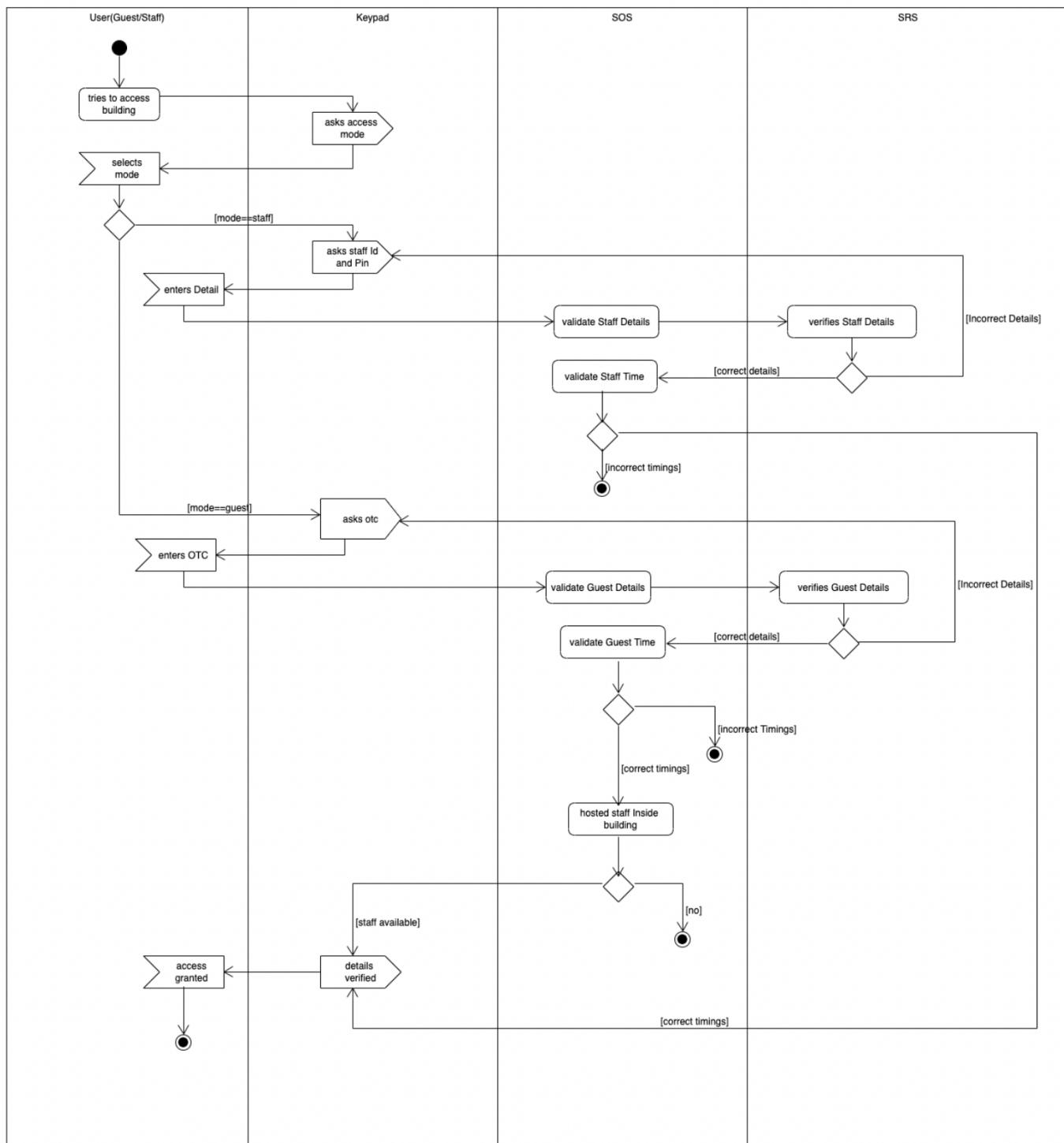
Staff – Main Flow Diagram



Staff – Alternative Flow Diagram

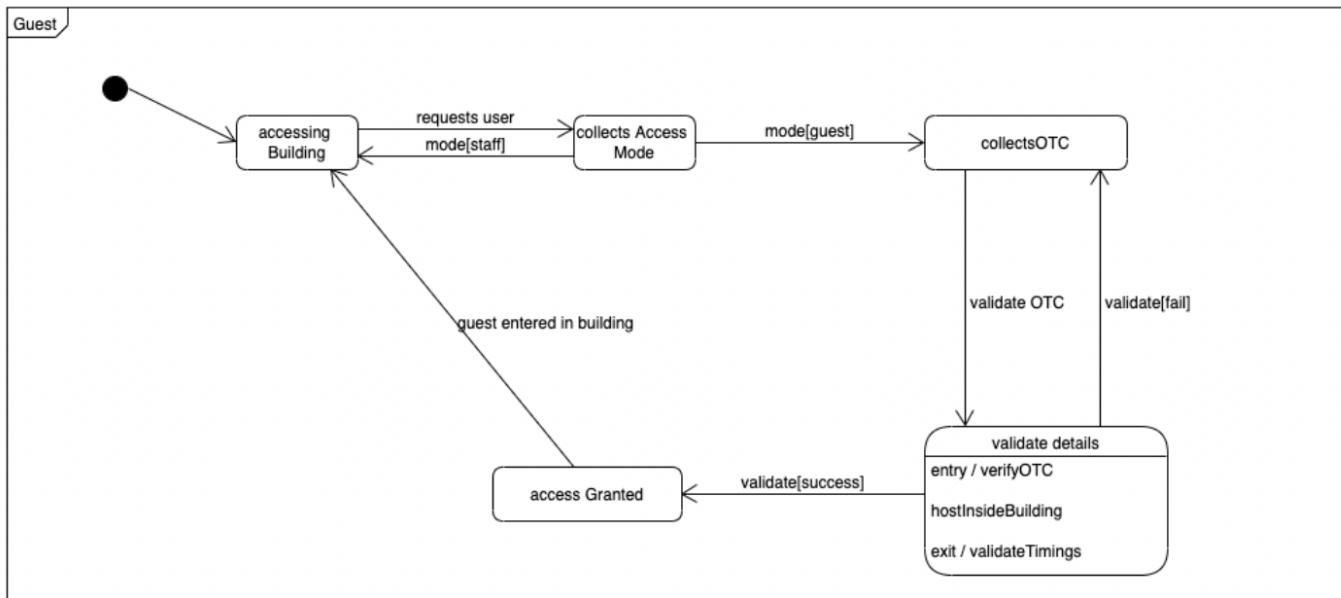


TASK 7 – ACTIVITY DIAGRAM

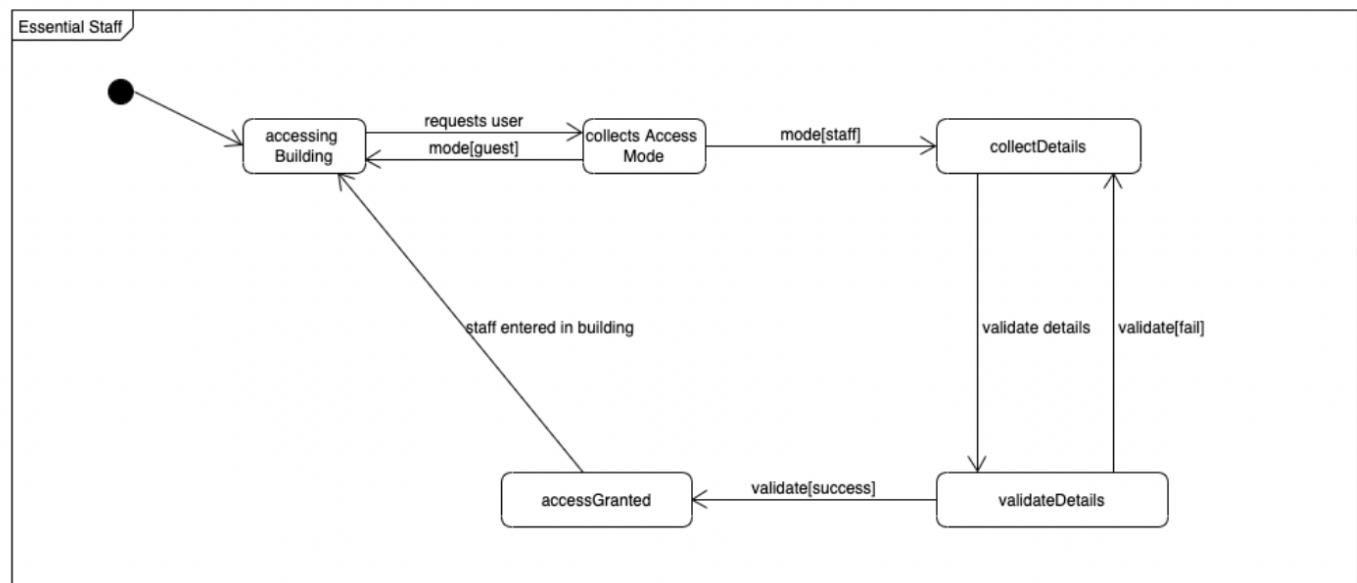


TASK 8 – STATE MACHINE DIAGRAMS

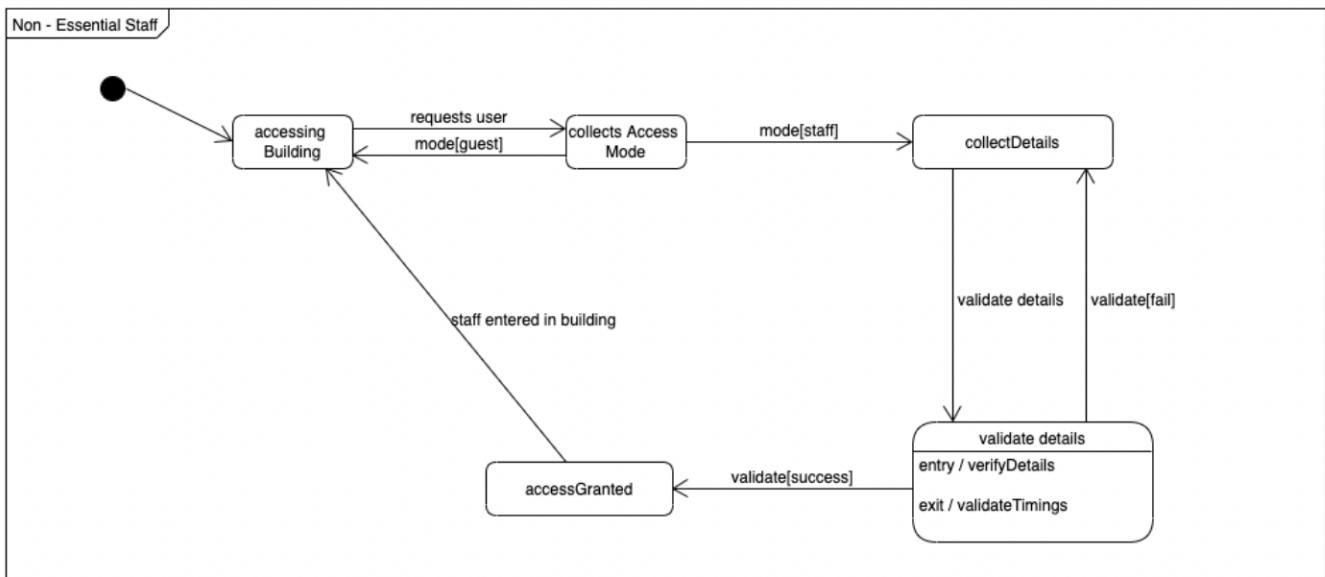
Guest – State Diagram



Essential Staff – State Diagram

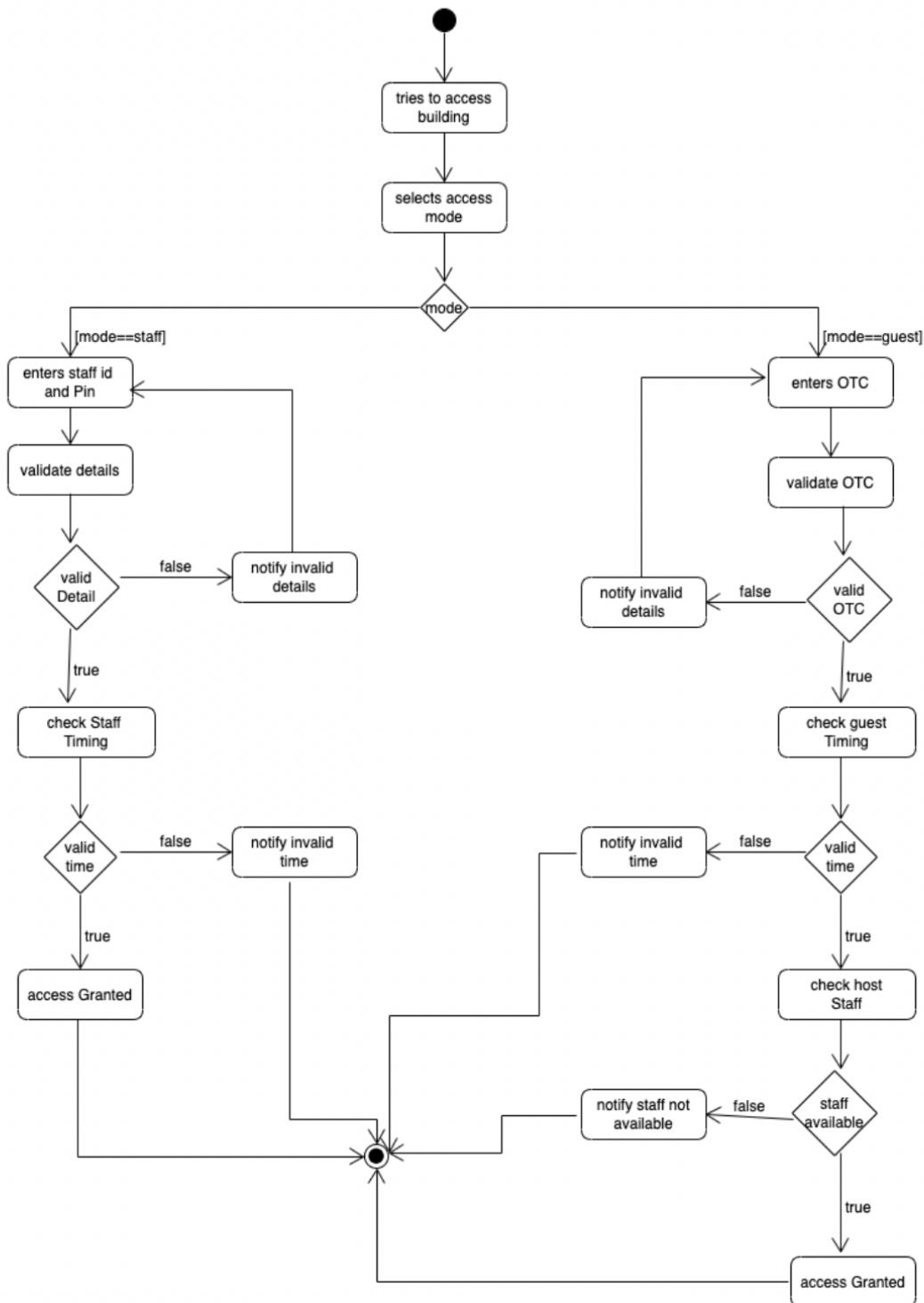


Non – Essential - State Diagram



TASK 9 – SCENARIO TEST CASES

Test Case Diagram



Round Trip Scenario Test Cases

| Path | Comment | Path Condition |
|------|--|---|
| 1 | Staff who enters valid staff Id and Pin and if current time is valid for that staff, then access is granted in first attempt. | <ul style="list-style-type: none"> • Selects StaffAccessMode • validateStaffDetails is true • validateStaffTime is true • accessGranted |
| 2 | Staff who enters invalid staff Id and Pin in first attempt and valid in second attempt and if current time is valid for that staff, then access is granted in second attempt. | <ul style="list-style-type: none"> • Selects StaffAccessMode • validateStaffDetails is false • validateStaffDetails is true • validateStaffTime is true • accessGranted |
| 3 | Staff who enters valid staff Id and Pin and if current time is invalid for that staff, then access is not granted. | <ul style="list-style-type: none"> • Selects StaffAccessMode • validateStaffDetails is true • validateStaffTime is false |
| 4 | Guest who enters valid OTC and if the current time is valid for that guest and if the host is already inside the building, then access is granted in the first attempt. | <ul style="list-style-type: none"> • Selects GuestAccessMode • validateOtc is true • validateGuestTime is true • checkHostInBuilding is true • accessGranted |
| 5 | Guest who enters invalid OTC in first attempt and valid in second attempt and if the current time is valid for that guest and if the host is already inside the building, then access is granted in the first attempt. | <ul style="list-style-type: none"> • Selects GuestAccessMode • validateOtc is false • validateOtc is true • validateGuestTime is true • checkHostInBuilding is true • accessGranted |
| 6 | Guest who enters valid OTC and if the current time is invalid for that guest, then access is not granted. | <ul style="list-style-type: none"> • Selects GuestAccessMode • validateOtc is true • validateGuestTime is false |
| 7 | Guest who enters valid OTC and if the current time is valid for that guest and if the host is not inside the building, then access is not granted. | <ul style="list-style-type: none"> • Selects GuestAccessMode • validateOtc is true • validateGuestTime is true • checkHostInBuilding is false |