Kingdom Of Saudi Arabia
Ministry Of Education
Taibah University
College Of Computer
Science and Engineering

المملكة العربية السعودية
وزارة التعليم
جامعة طيبة
كلية علوم وهندسة الحاسب الآلي

Embassy system

A Project Report Submitted to Fulfill the Requirements of CS284

By Group (8):

Waseem Zahr Aljumah (SID: 4206111)
Yaseen Ahmad Albar (SID: 4201503)

Supervised By

Dr. Mohammad M. Alsuraihi

5 May, 2024

TABLE OF CONTENTS

1 Introduction

1.1 Problem definition

It was noted in previous years that a person needs time to access embassy services, so he goes to the embassy to be able to request a visa, passport, or other service. This takes time, and also causes congestion in the embassy due to the large number of people wishing to request services, which leads to the inability of employees to finish. From all people in record time, and it is possible that some paper transactions may be damaged or lost due to their large number.

1.2 Proposed solution

A system will be created that will enable people to request services through the system without the need to go to the embassy, so that people can request a passport or visa through the system. This will make it easier for people to request services and will also make it easier for employees to receive requests, verify them, and preserve them from damage or damage. loss, and the system will contain services that benefit both employees and people.

2 Requirements Elicitation

2.1 Interviews

Developer: What system do you need?

Client: I want a system for the embassy.

Developer: Do you want an application system or a standalone application system?

Client: I want a standalone application system.

Developer: Who are the users who will use the system?

Client: I want there to be 3 users, namely Employee, User, and Guest.

Developer: What can an EMPLOYEE do?

Client: He must log in using his NAFATH account and can view visa applications (and can accept or reject applications), passport applications (and can accept or reject applications) add clubs, show clubs, view contact information, and view embassy location.

Developer: What can the USER do?

Client: Logs in using his NAFATH account, can submit an application for a visa and pays the required amount, and can submit an application for a passport and pays the required amount, Can show clubs, view contact information, and view embassy location.

Developer: What can the GUEST do?

Client: Can show clubs, view contact information, and view embassy location.

Developer: What non-functional functionality are you interested in providing for the app?

Client: It must be safe from hacking, its security and protection are very high, it must be available at all times, it must perform its functions reliably and flawlessly, it must be easy to use and uncomplicated in order to be popular with users and employees, and it must be maintainable. And modify easily.

2.2 Scenarios

1- The user logs in using his NAFATH account, and then the system verifies the validity of the entered ID and password through the database of the NAFATH system, then the user requests the issuance of a visa, then he makes the payment process and can cancel the payment when access the payment stage and must have sufficient balance for the payment to be completed successfully.

2- The employee logs in using his NAFATH account, and then the system verifies the validity of the ID entered and the password entered through the database of the NAFATH system displays the visa applications, and sees whether the requirements are complete or not. If they are complete, the application for issuing a visa is accepted. If it is incomplete, the visa application will be rejected.

2.3 Definition requirements

2.3.1 Functional requirements

The employee must log in using a NAFATH account to access the system.

The employee can view visa applications and accept or reject the application.

The employee can view passport applications and accept or reject the application.

The employee can add clubs.

The user must log in using a NAFATH account to access system services.

The user can request the visa and pay the required amount.

The user can request the passport and pay the required amount.

Guest, employees, and users can view the clubs.

Guest, employees, and users can view contact information.

Guests, employees, and users can view the embassy website.

2.3.2 Nonfunctional requirements

Nonfunctional requirements	Explanation
SAFETY	The system will be secure against any hacking that will happen to the system.
AVAILABILITY	The system must be available all the time.
RELIABILITY	The software framework reliably and flawlessly executes the listed functions.
USABILITY	The simplicity at which a user can communicate with a machine to read, run, plan inputs, and interpret outputs.
MAINTAINABILITY	The ability to perform maintenance on the system, such as adding other functions to the system or modifying it.

3 System Analysis Models

3.1 Description of system models

To build this system, we will use the model in the analysis stage, which is the functional model, object model, and dynamic model.

- **Functional Model:**

Use Case Diagrams: Showcase different situations and user interactions with the system.

Importance: Functional diagrams offer a thorough comprehension of the behavior of the system as seen by the user. They help extract requirements, define expectations, and make sure the system serves the intended function. Functional diagrams provide a foundation for future development and verification by detailing the functionalities of the system and how users interact with it.

- **Object Model:**

Class Diagrams: Explain the links between items and their structures.

Importance: Object diagrams help in system architecture design and functional implementation. Object diagrams help with modularization, encapsulation, and code reuse by depicting system components as objects and specifying their properties and actions. They divide the system into manageable components, allowing developers to manage complexity and upkeep.

- **Dynamic Model:**

Sequence Diagrams: Show the message flow that occurs between objects when the system is operating.

State Chart Diagrams: Explain the states and transitions of objects in relation to events.

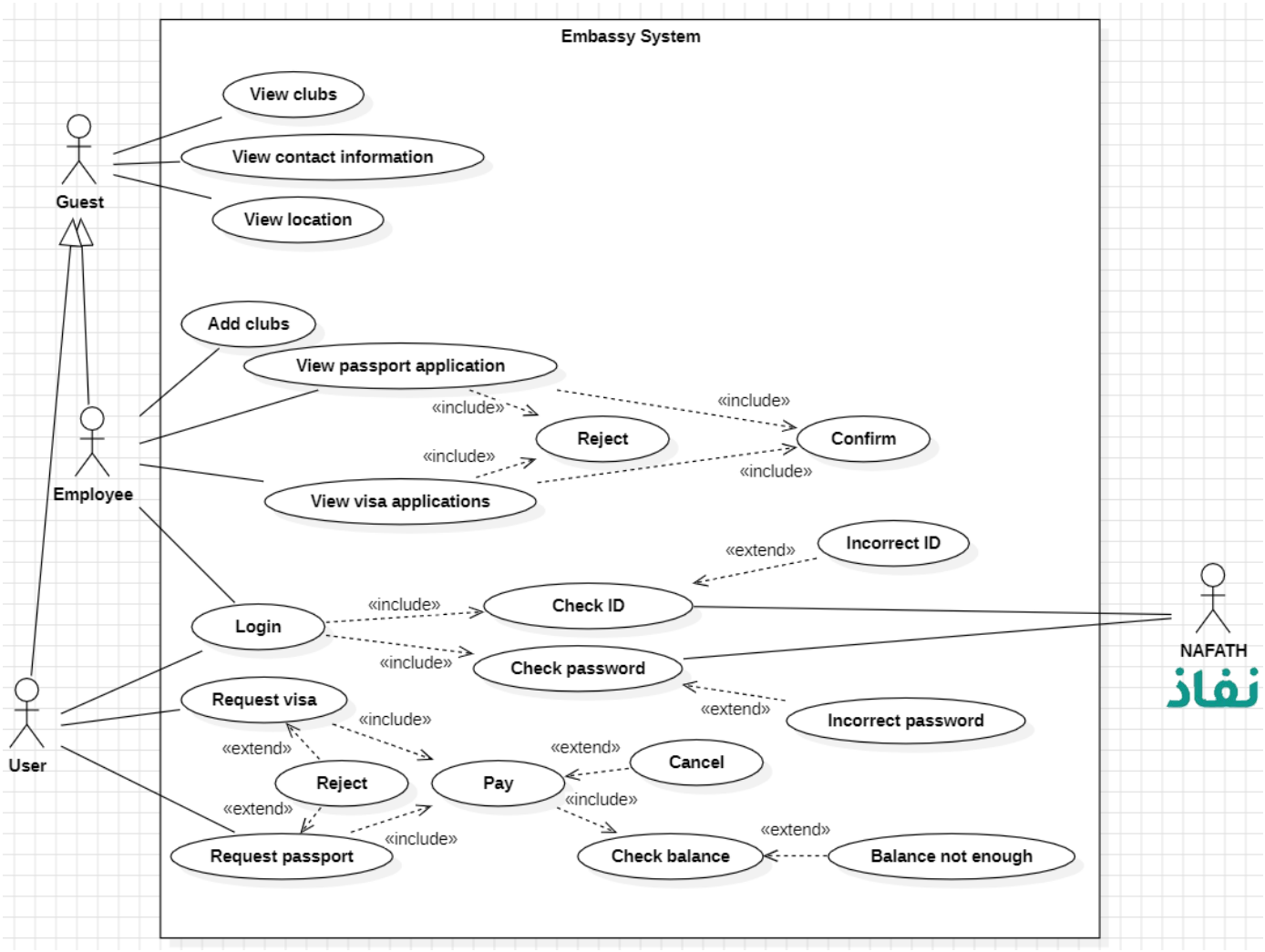
Activity Diagrams: Show how the system's operations go, including choices, actions, and concurrency.

Importance: The behavior of the system during runtime may be understood by using dynamic diagrams, which can also be used to spot possible problems including concurrency difficulties, race situations, and performance bottlenecks. Dynamic diagrams help to confirm system correctness, identify design problems, and optimize system performance by showing interactions between objects, activity sequences, and the flow of control.

3.2 System models

3.2.1 Functional model

3.2.1.1 Use Case Diagram



Description: This model is the Use Case Diagram, and it contains the functions that the system will provide and those who have the authority to access these functions. It explains the relationships between the functions and the users. We have Extended and Included relationships between the Use Cases and Generalization between the actors.

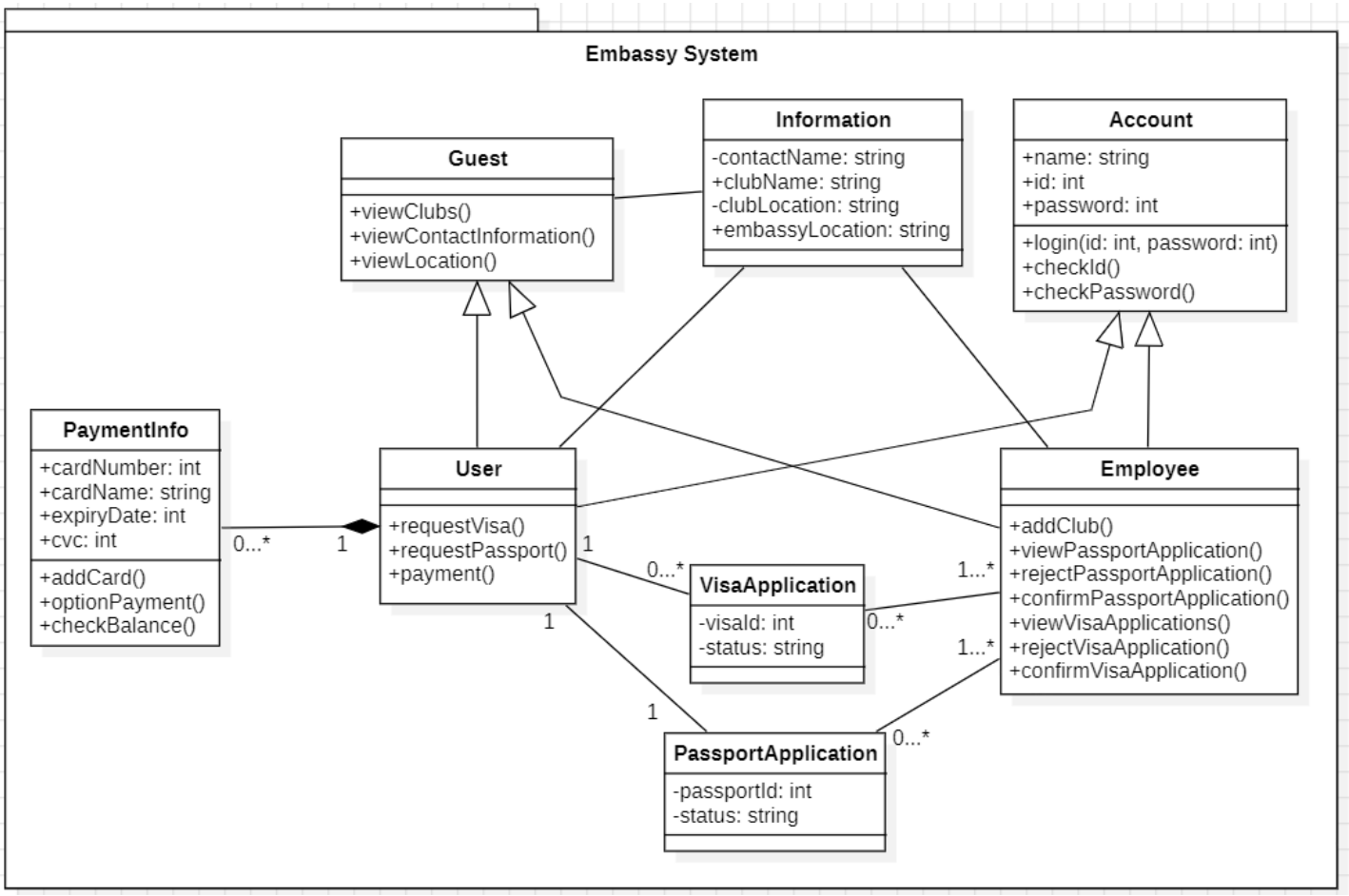
3.2.1.2 Textual

Name	Request visa/passport
Participating actor	User
Entry condition	The user log in using NAFATH account. The user has balance enough.
Exit condition	User has obtained a visa/passport.
Flow of events	1- The user logs in using NAFATH account. 2- The user request for a visa/passport. 3- The employee reviews the user's application for a visa/passport. 4- Upon acceptance, the amount due is displayed to the user. 5- The user makes a payment of the specified amount. 6- The system issues the visa/passport.
Special requirement	

Name	Log in
Participating actor	Employee, User
Entry condition	The user or employee enters NAFATH account information correctly.
Exit condition	The user or employee is logged in.
Flow of events	1- The user or employee log in through his account in NAFATH. 2- The system verifies the validity of the entered ID. 3- The system verifies the validity of the entered password.
Special requirement	Verifying the validity of the account data entered from the NAFATH.

3.2.2 Object model

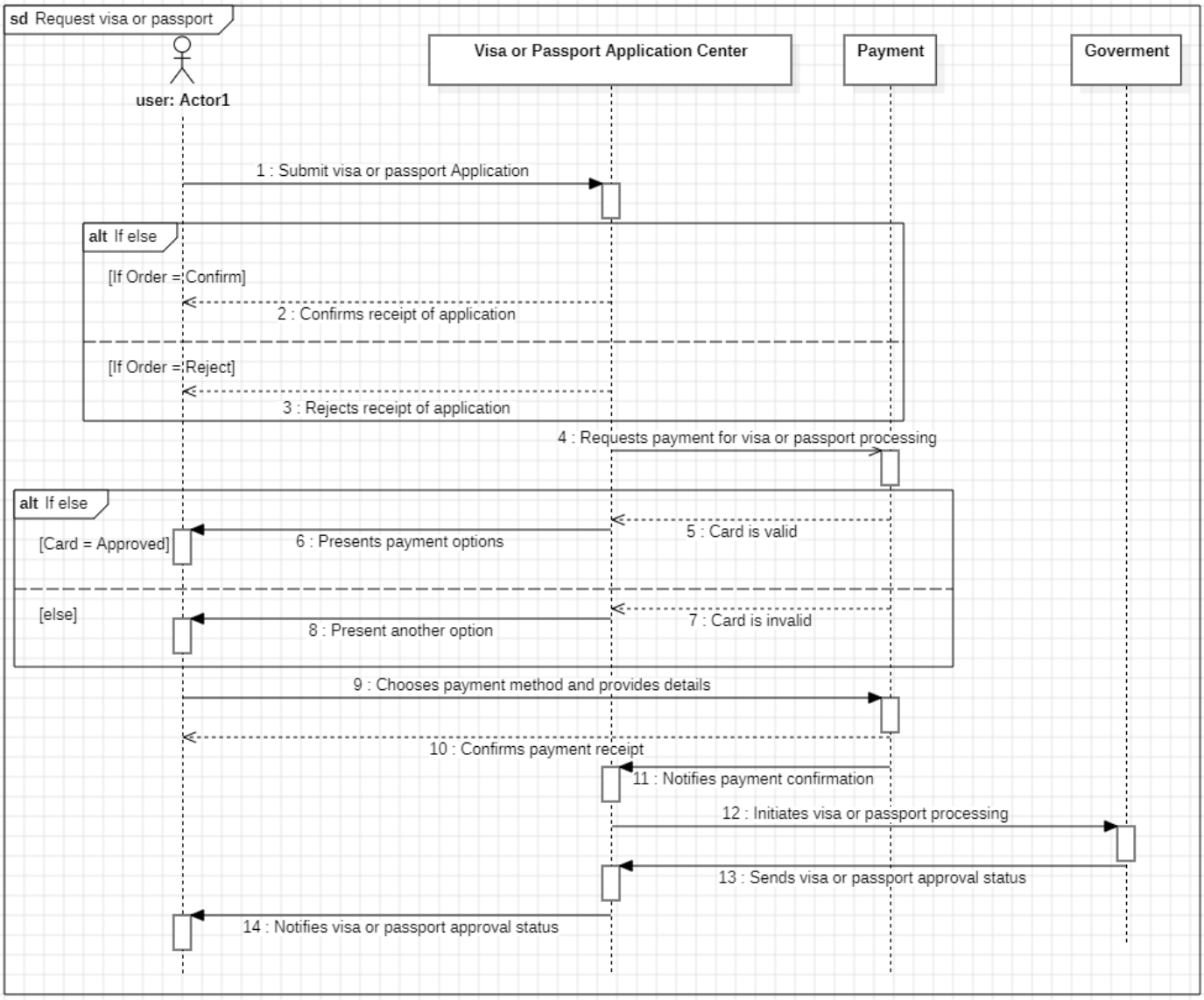
3.2.2.1 Class Diagram



Description: This model is the Diagram class, and there are divisions of the classes. Each class contains features and characteristics, and there are relationships between the classes. One of these relationships is the relationship of the user class to the payment class, and the relationship between them is called Composition. A single user can make unlimited payment operations, and he can also not make any payment process. We also have a relationship between the User and Employee classes and the Account class, and the relationship between them is inheritance, which means that the User and Employee classes inherit the features and characteristics of the Account class.

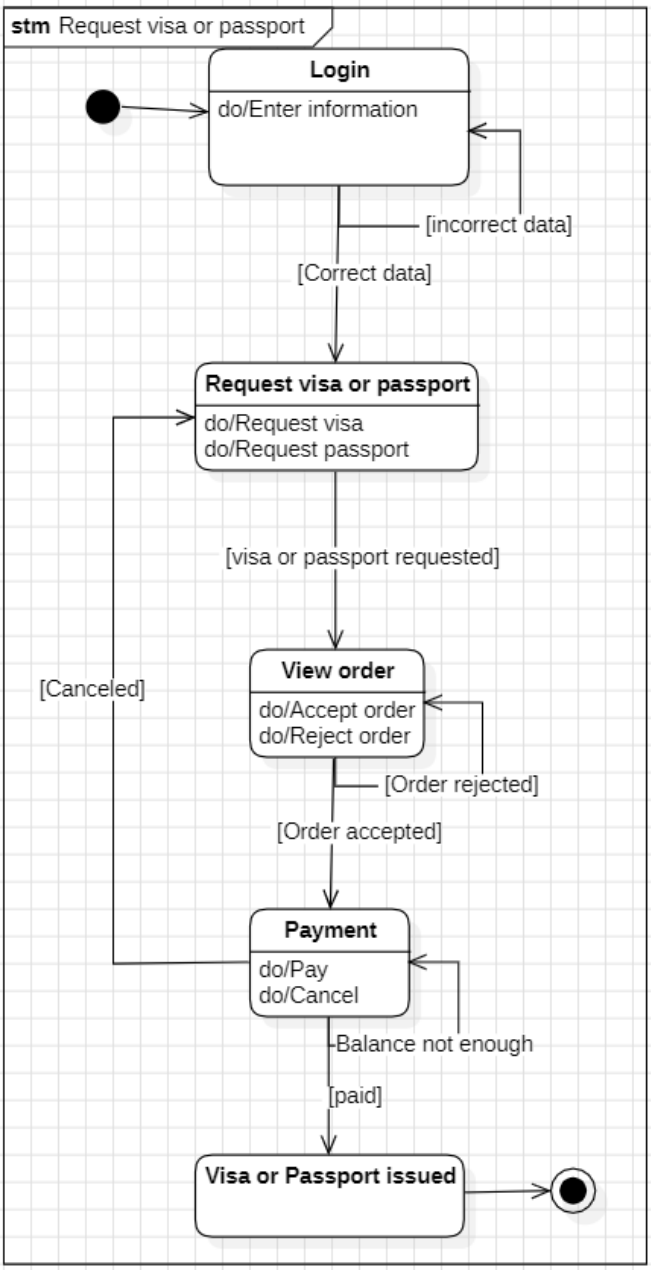
3.2.3 Dynamic model

3.2.3.1 Sequence Diagram



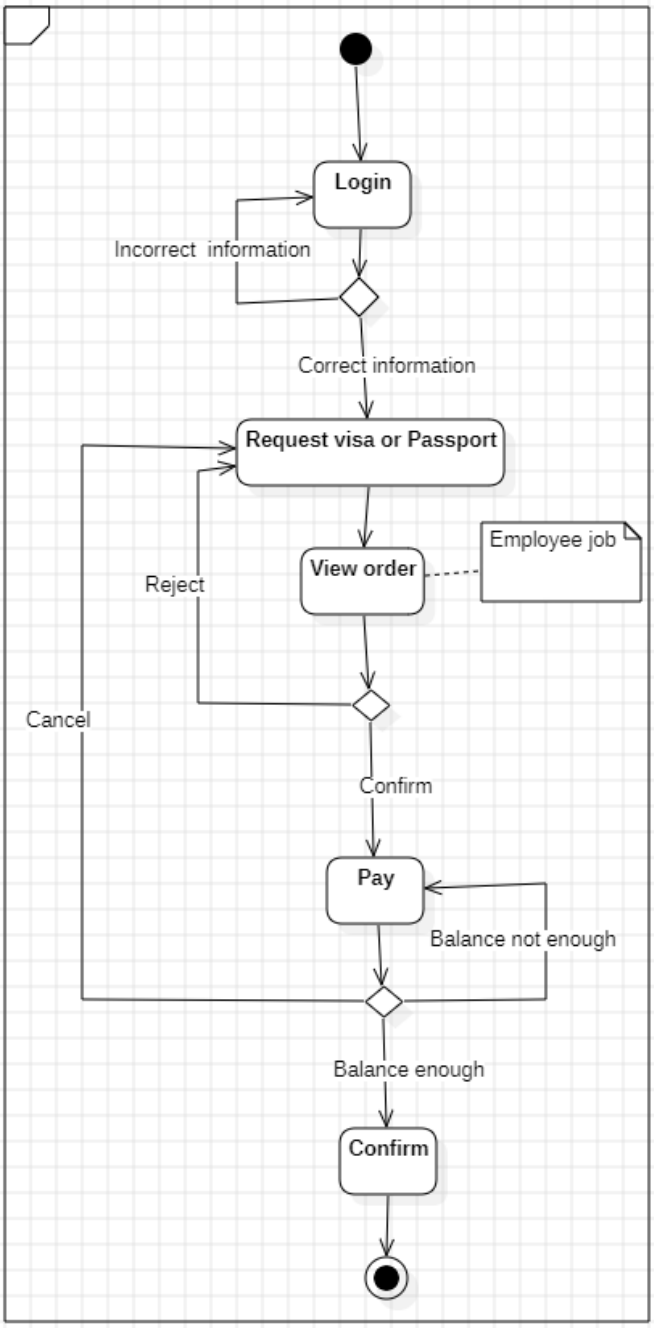
Description: In this model, the sequence of processes for applying for a visa/passport and the steps that will be carried out from the beginning of the process to its end are explained sequentially.

3.2.3.2 State Chart Diagram



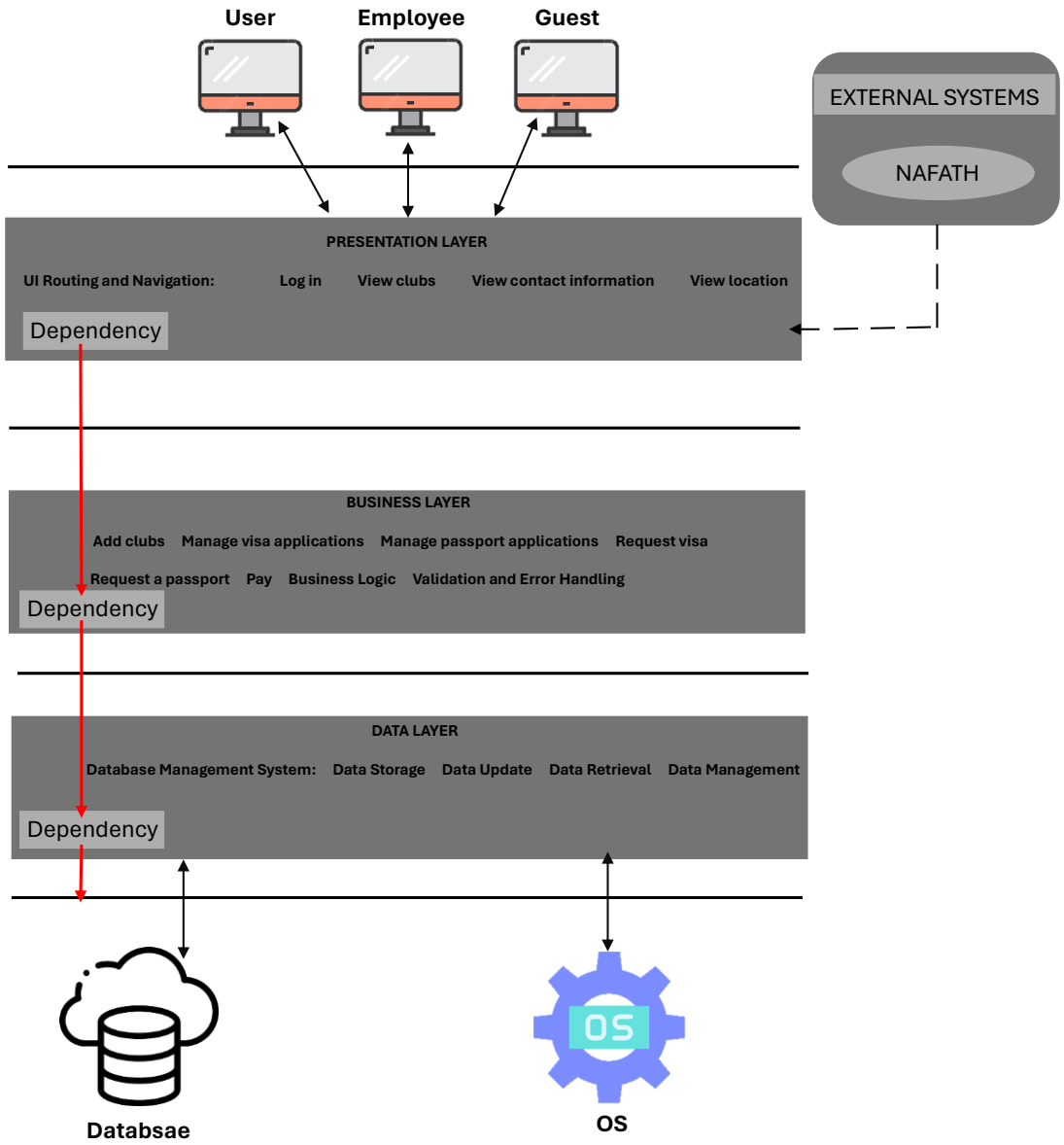
Description: In this model, the operations that are in one object that will be implemented in each state and the condition that is transferred from one state to another are described.

3.2.3.3 Activity Diagram



Description: In this drawing, we have explained the sequence of activities for the visa/Passport application process, as it begins with a log in and the information entered must be correct, then the user chooses to request a visa, and then the employee displays the application. If the employee approves the request, he moves to the payment stage, and when paying, the balance must be sufficient. To complete the process, after payment, the order is completed.

4 System Architectural Design



Description: This architectural design is layered and was built in 4 layers.

This type helps to enhance security, as it prevents access to the data directly, as the layers are passed one after the other until the database is reached, but the system user cannot access it and know its contents because of the layers that precede the database.

5 Summary

A program was worked on that provides embassy services, as it provides services that make it easy for the user to request them, save time for employees, and help them reduce errors. An interview was conducted with the embassy's client to find out the services that the system will provide, its working mechanism, and their needs. Models were built that explain the structure of the system from all points of view. Aspects and these models include the functional model (Use Case Diagram), the Object model (Class Diagram), Dynamic model (Sequence Diagram, State Chart Diagram, and Activity Diagram), which makes it easier for the developer who will write the codes for the system to understand the system, its characteristics, and functions, The structure of the system was created, which shows how it will be divided and how it will be connected. An architectural design is layered and was built in 4 layers, and this design enhances and raises security so that it preserves user data from theft and spread and enhances protection for the system to prevent unethical hacks and violations.

The system can be developed in the future and other services can be added to it, such as: adding a section dedicated to news, a section dedicated to events, and some other functional services, such as: requesting a lawyer, requesting a visa extension, and requesting the issuance of a birth certificate for those born abroad, and print visa/passport, as this will increase the accreditation rate. Therefore, errors, if any, can be improved and security strengthened.