Palestine Technical University-Kadoorie

Faculty of Engineering and Technology

Computer Systems Engineering Department

Information Systems Analysis and Design

"Implementing a Palestinian Tabou Website"

Prepared By:

Mays Hawwa – 201910360

Mohammad Eneirat – 201911370

Osama Marei – 201911195

Supervised by:

Dr. Yazeed Sleet

Tulkarm, Palestine
May, 2023

This document was submitted in fulfilment of the requirements for the ISAD (Information Systems Analysis & Design) curriculum, during the academic year 2022-2023 (2nd Semester).

1. Use Case Diagrams

1.1 Admin Use Case Diagram

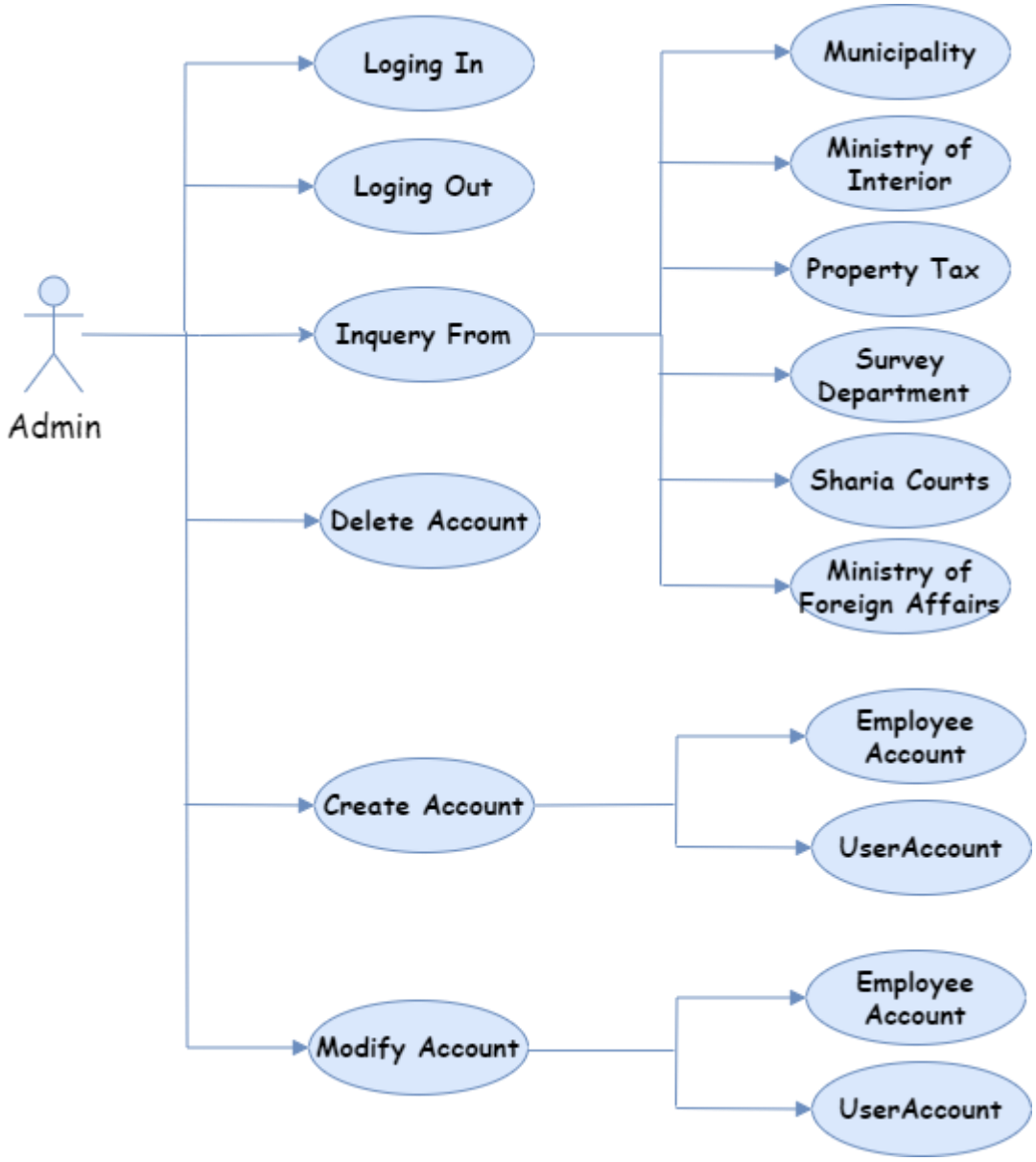


Figure 1: Admin Use Case

1.2 Employee Use Case Diagram

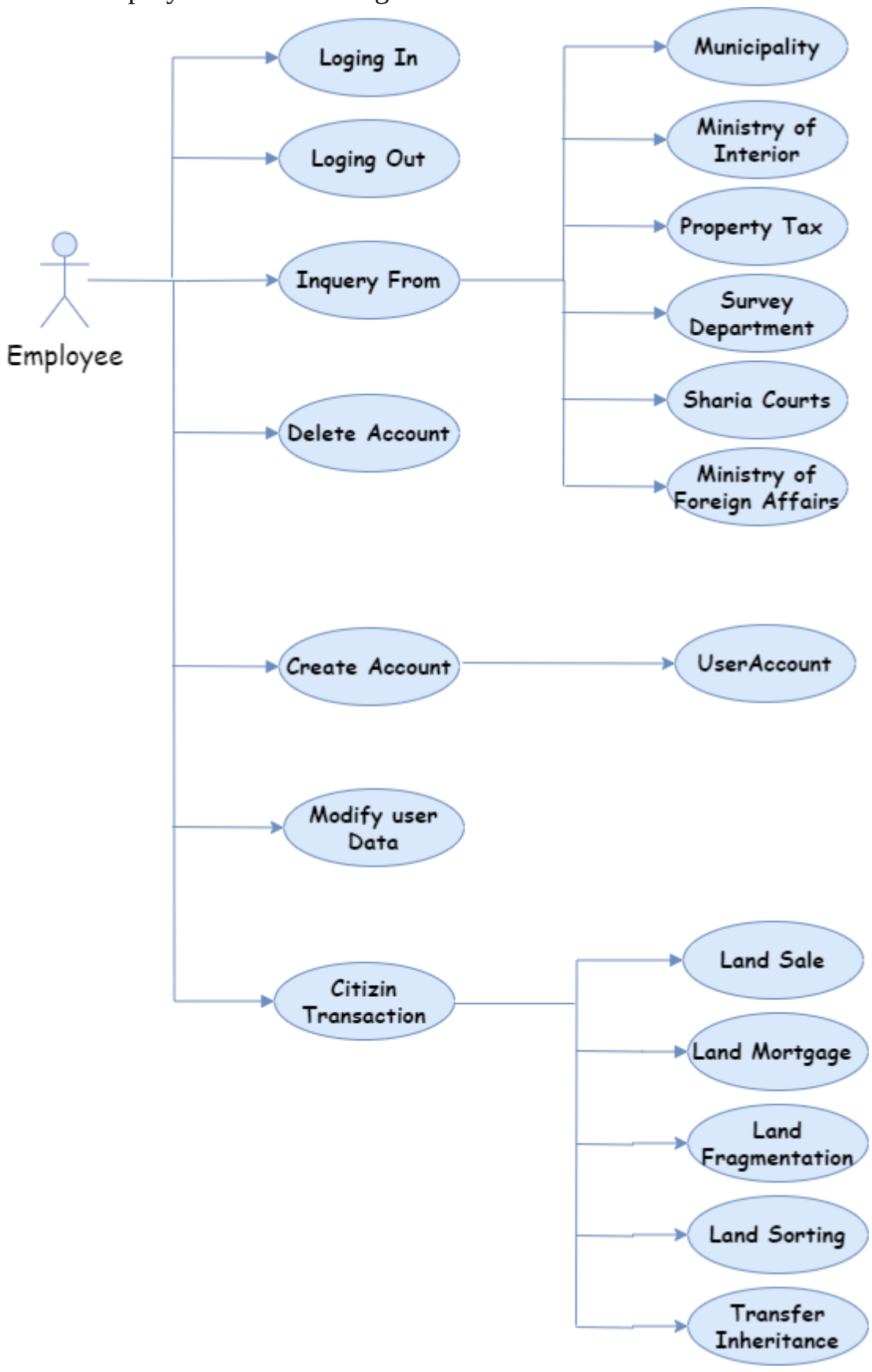


Figure 2: Employee Use Case

1.3 User Use Case Diagram

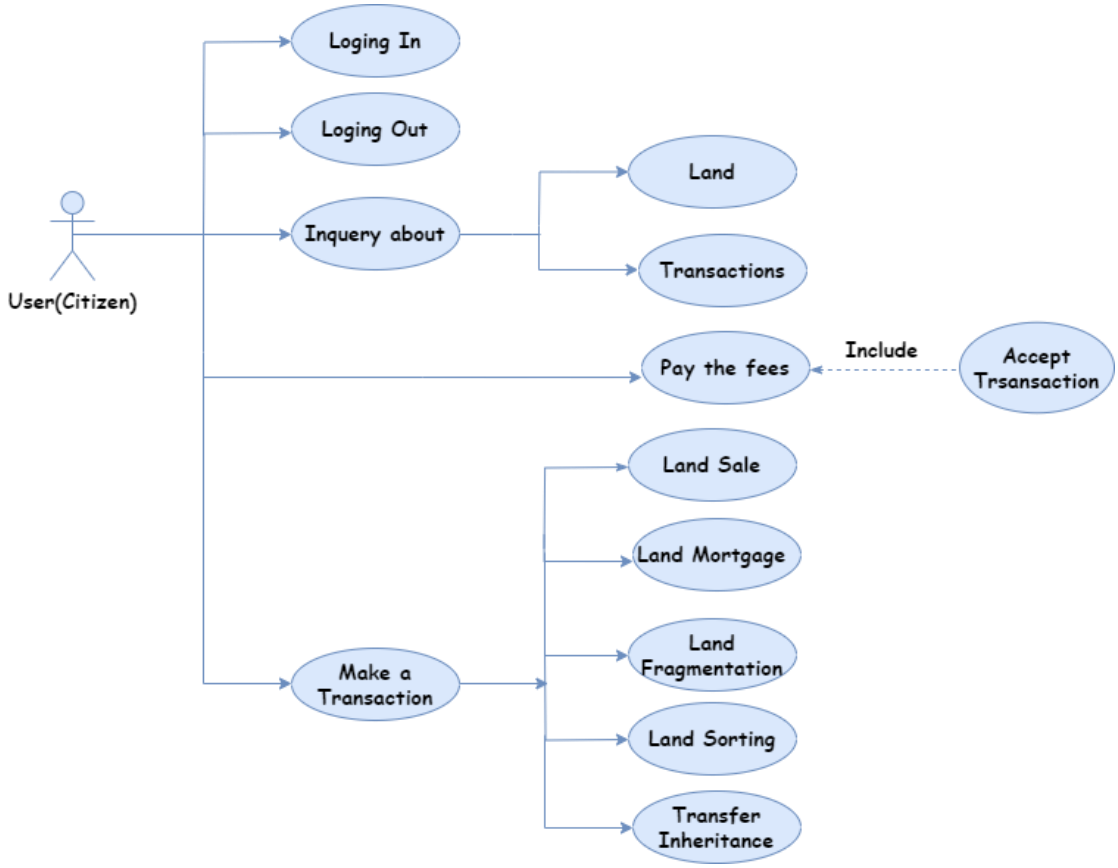


Figure 3: User Use Case

2. Entity Relationship Diagram.

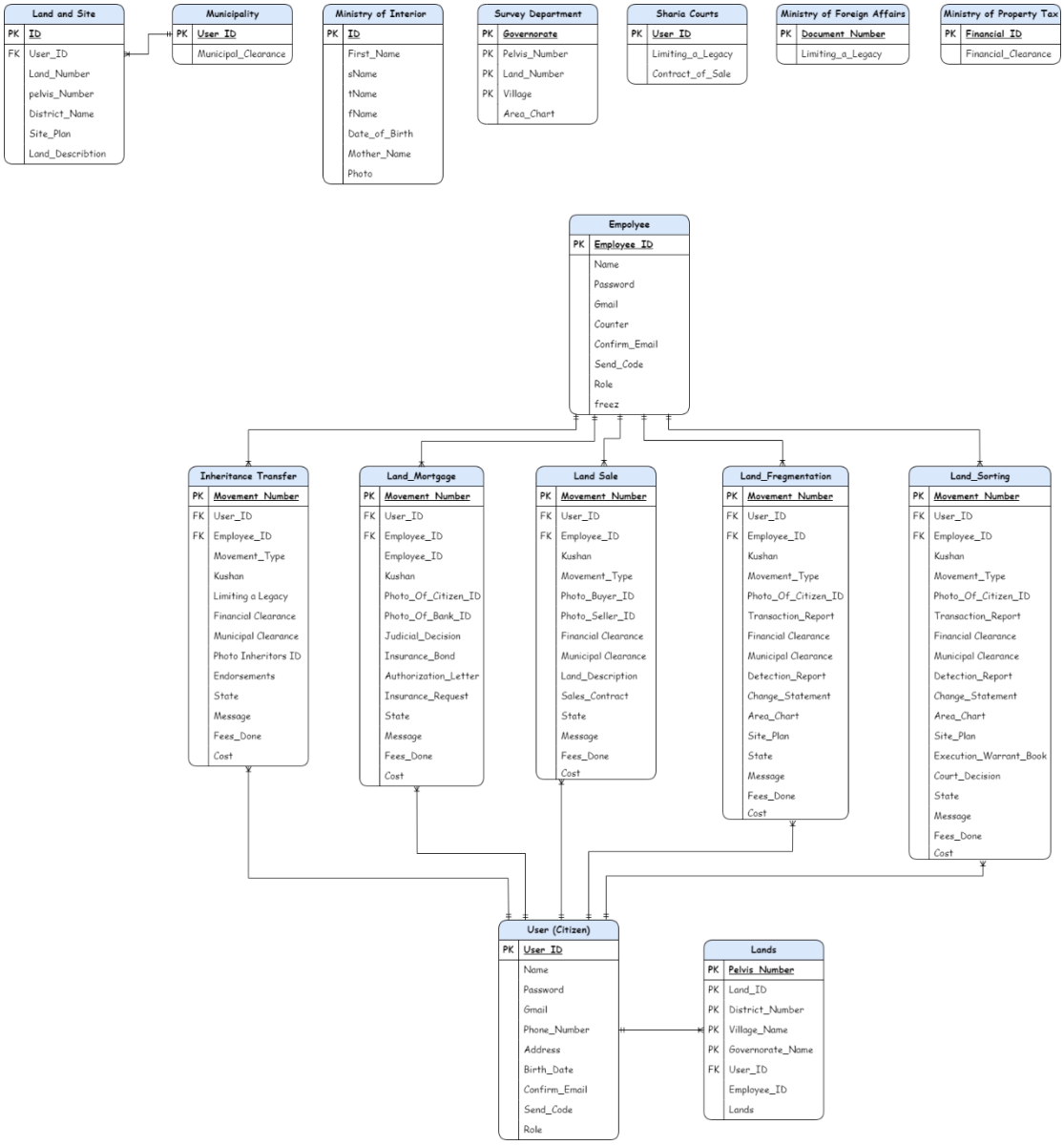


Figure 4: ERD

3. Domain Model Class Diagram

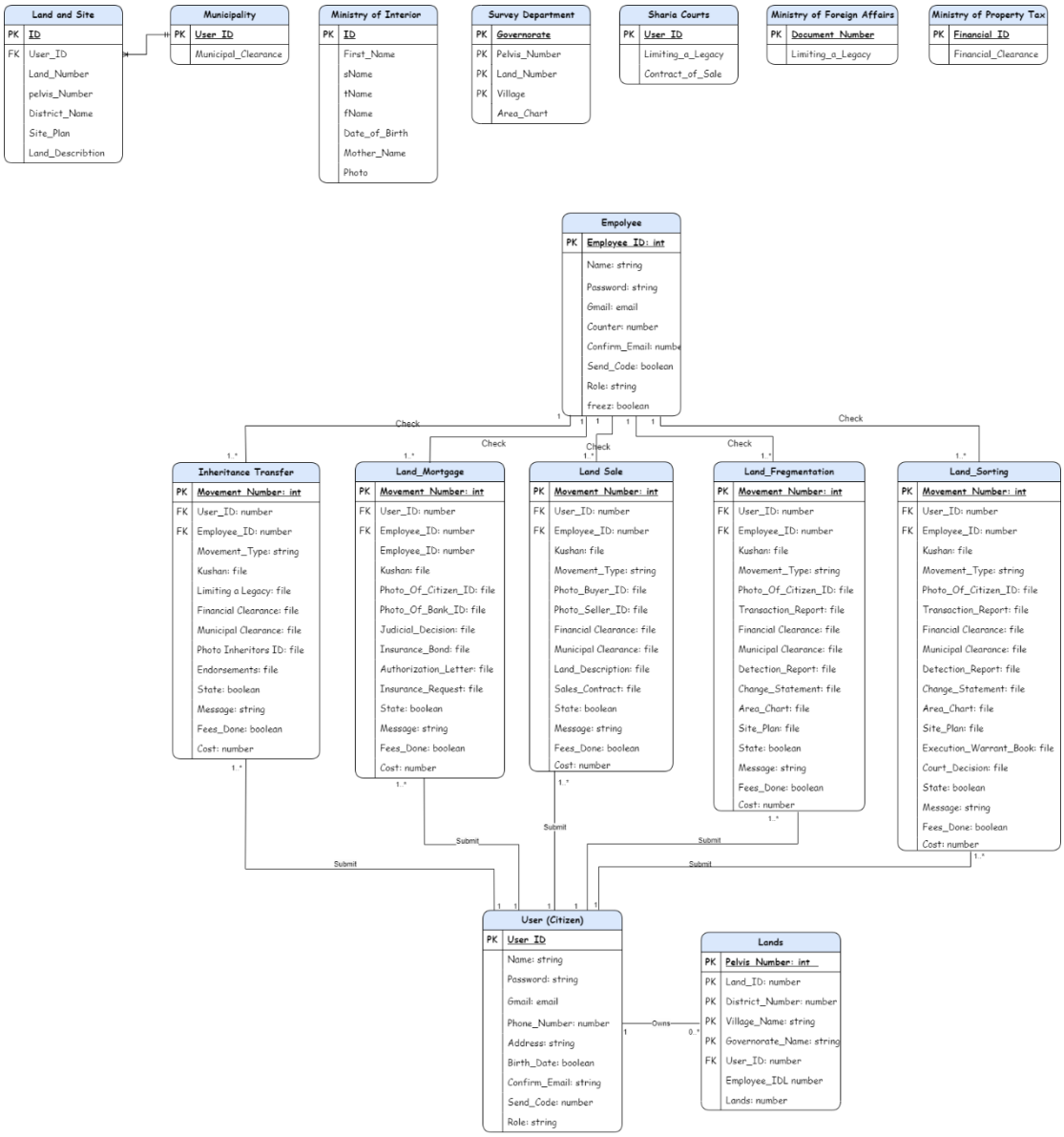


Figure 5: Domain Model Class Diagram

4. Fully Developed Use Case Description

4.1 Log-In full use case Description

Table 1: Log-In Full use case Desription

UseCase Name:	Sign-In	
Scenario:	Sign into the system using a predefined account.	
Triggering event:	User requests to Sign In.	
Brief description:	This use case allows user to sign in into the system to access the relevant functions according to the user's role. The various user roles are admin, employee and user. To sign in to the system, all users have to enter their unique user email. Upon successful sign in the system will display the relevant user's home page.	
Actors:	Admin, Employee and User	
Related use cases:	No related use cases	
Stakeholders:	Citizen and employees and Admin	
Preconditions:	A user account is existed in the database. The user is not already logged in. A validation and authentication must be available	
Postconditions:	1. The user is logged in to the system. 2. The user has access to the functions of the system.	
Flow of activities:	Actor:	System:
	The actor enters his/her email and password	1. The system requests that the actor enter his/her email and password. 2. The system validates the entered email and password and logs the actor into the system.
Exception conditions:	Invalid email / Password If in the Basic Flow the actor enters an invalid email and/or password, the system displays an error message. The actor can choose to either return to the beginning of the Basic Flow or cancel the login, at which point the use case ends	

4.2 Create User Account

Table2 : Create User Account full use case Description

UseCase Name:	Create User	
Scenario:	The employee wants to create a user account.	
Triggering event:	The employee logging in his account, clicks on "Create New User."	
Brief description:	This use case allows emloyee to create new accounts to have access to the system.	
Actors:	employee.	
Related use cases:	No related use cases	
Stakeholders:	Employees	
Preconditions:	1. The employee is logged in 2. The new user email and id is be unique.	
Postconditions:	1. Success: The User entered data is stored in the user account, a confirmation is sent to the entered email.	
Flow of activities:	Actor:	System:
	The employee enters the user's username, password, email, address,birthdate, ID and phone number	1. The system prompts the employee for the user's username, password, email, address, ID, birthDate and phone number, The User is created. 2. The system displays a message indicating that the user is created.
Exception conditions:	The user account was not created: The Employee entered invalid user data, or chose to cancel the account creation request. In either case no account will be created.	

4.3 Create Employee Account

Table 3: Create Employee Account full use case Description

UseCase Name:	Create Employee Account

Scenario:	The admin wants to create an employee account.	
Triggering event:	The admin clicks on “Create New Employee”.	
Brief description:	This use case allows admin to create new accounts to have access to the system.	
Actors:	Admin.	
Related use cases:	No related use cases	
Stakeholders:	Admins	
Preconditions:	1. The admin is logged in 2. The new employee email and id is unique.	
Postconditions:	1.Success: The User entered data is stored in the user account, a confirmation is sent to the appropriate email address.	
Flow of activities:	Actor:	System:
	The admin enters the employee’s username, password, email, address, ID, address and phone number	1. The system prompts the employee for the user’s username, password, email and ID, the employee account is created. 2. The system displays a message indicating that the employee account is created.
Exception conditions:	The employee account was not created: The Admin entered invalid employee data, or chose to cancel the account creation request. In either case no account will be created.	

5. Activity Diagram.

5.1 Admin Activity

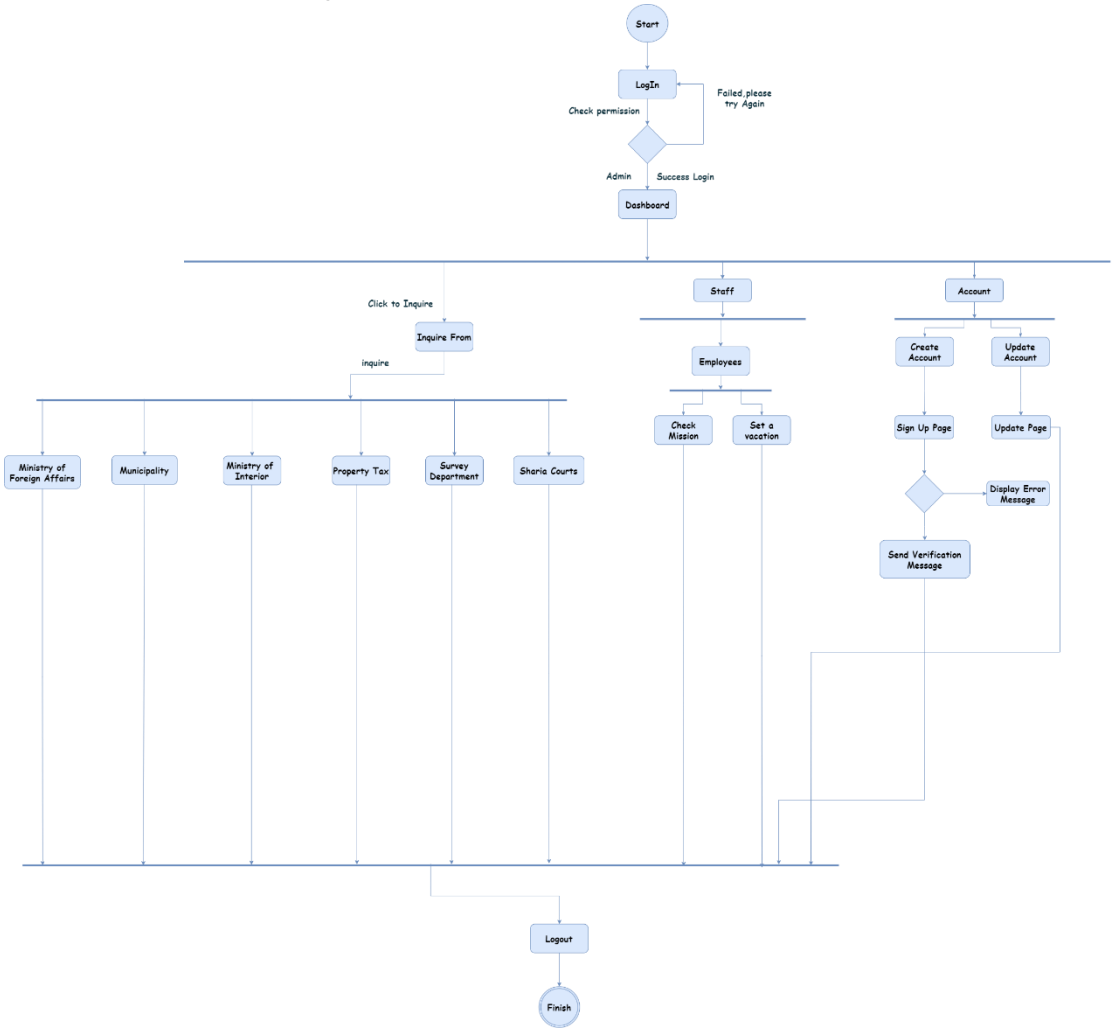


Figure 6: : Admin Activity Diagram

5.2 Employee Activity

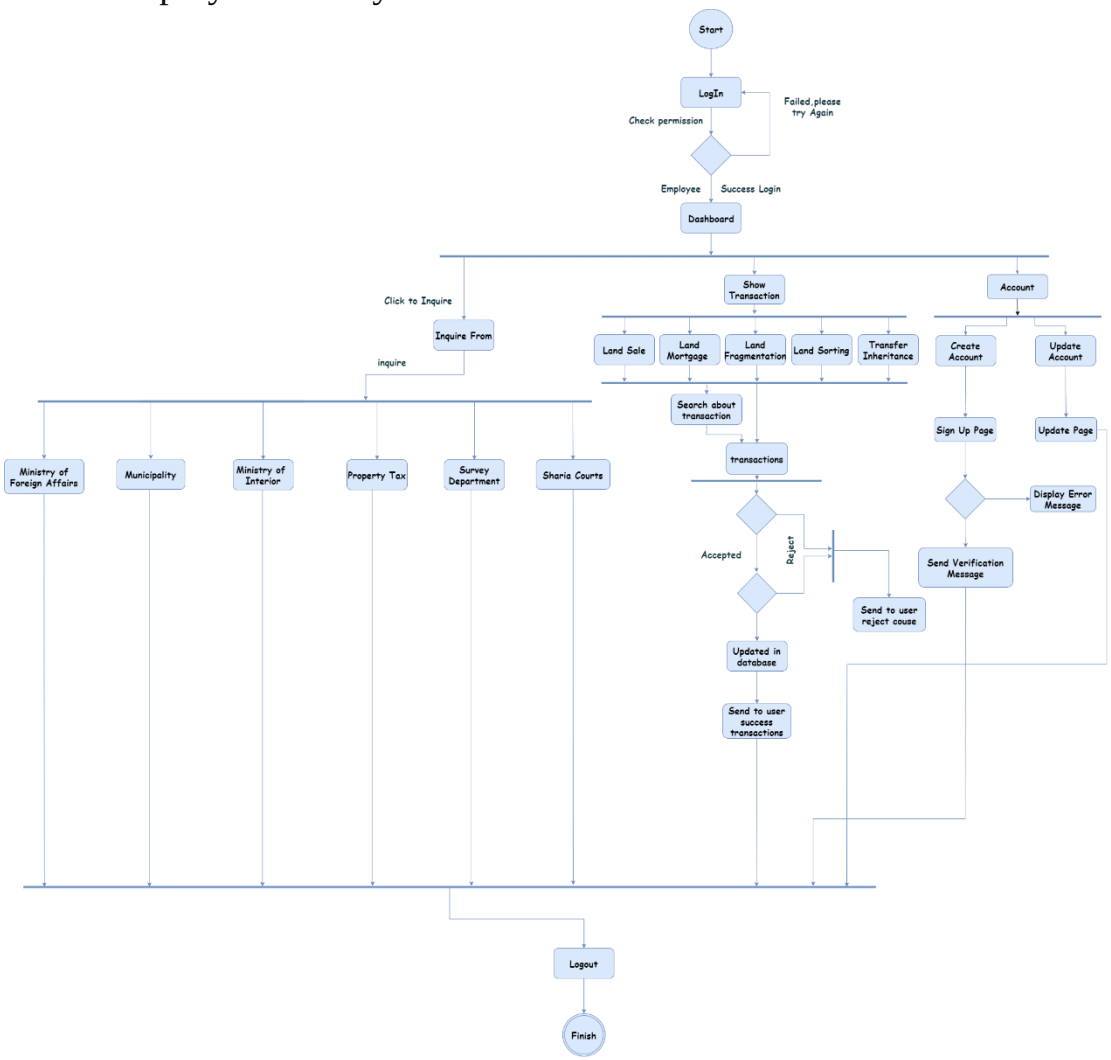


Figure 7: Employee Activity Diagram

5.3 User Activity

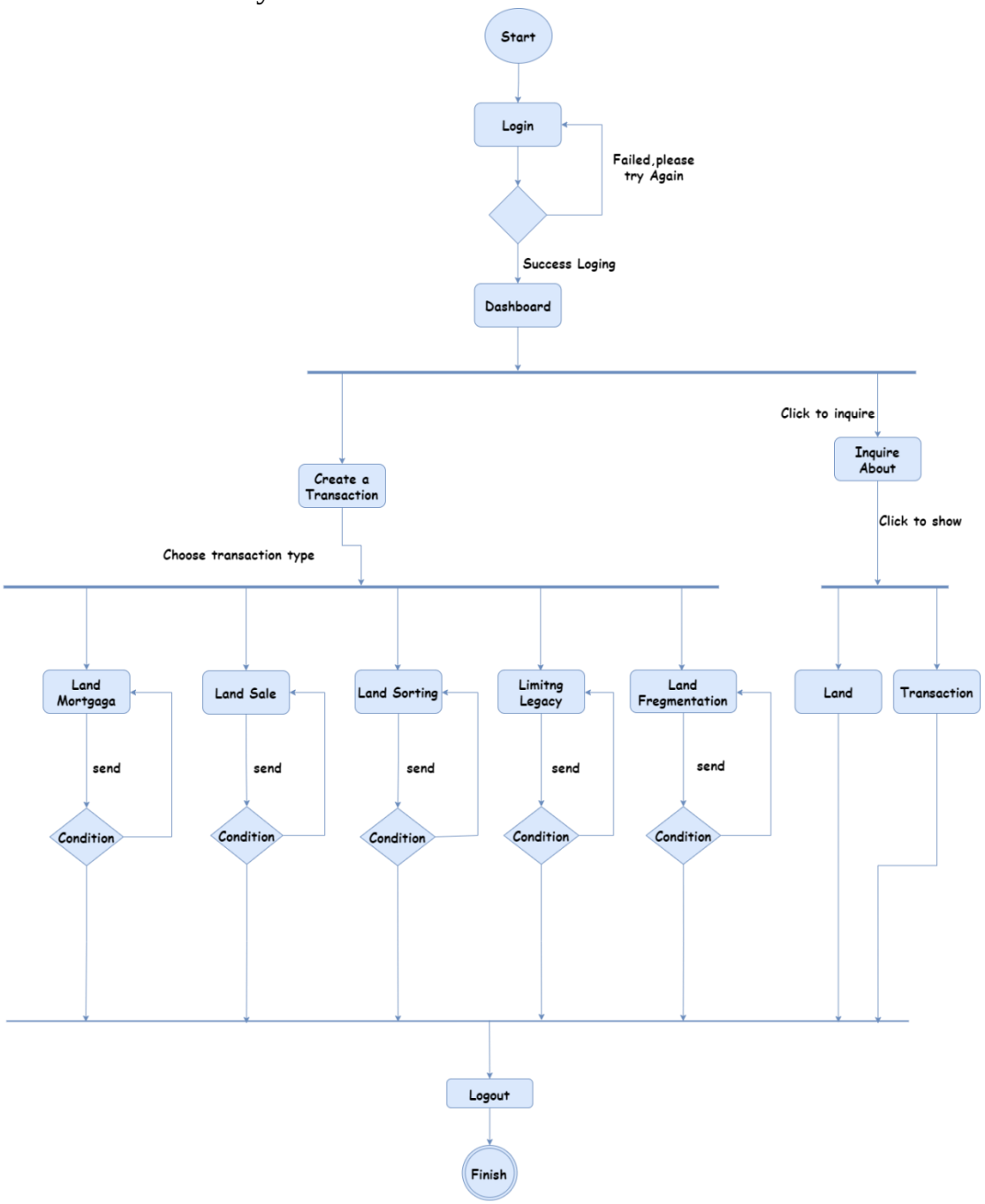


Figure 8: User Activity Diagram

6. System Sequence Diagram

6.1 Log-In

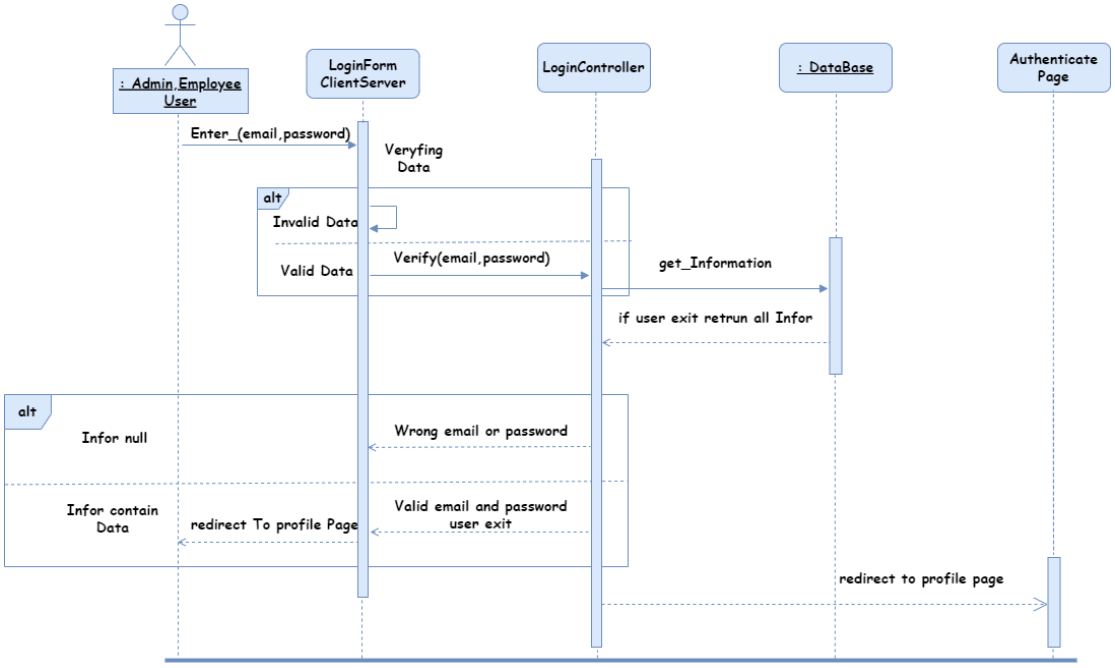


Figure 9: Log-In system Sequence

6.2 User

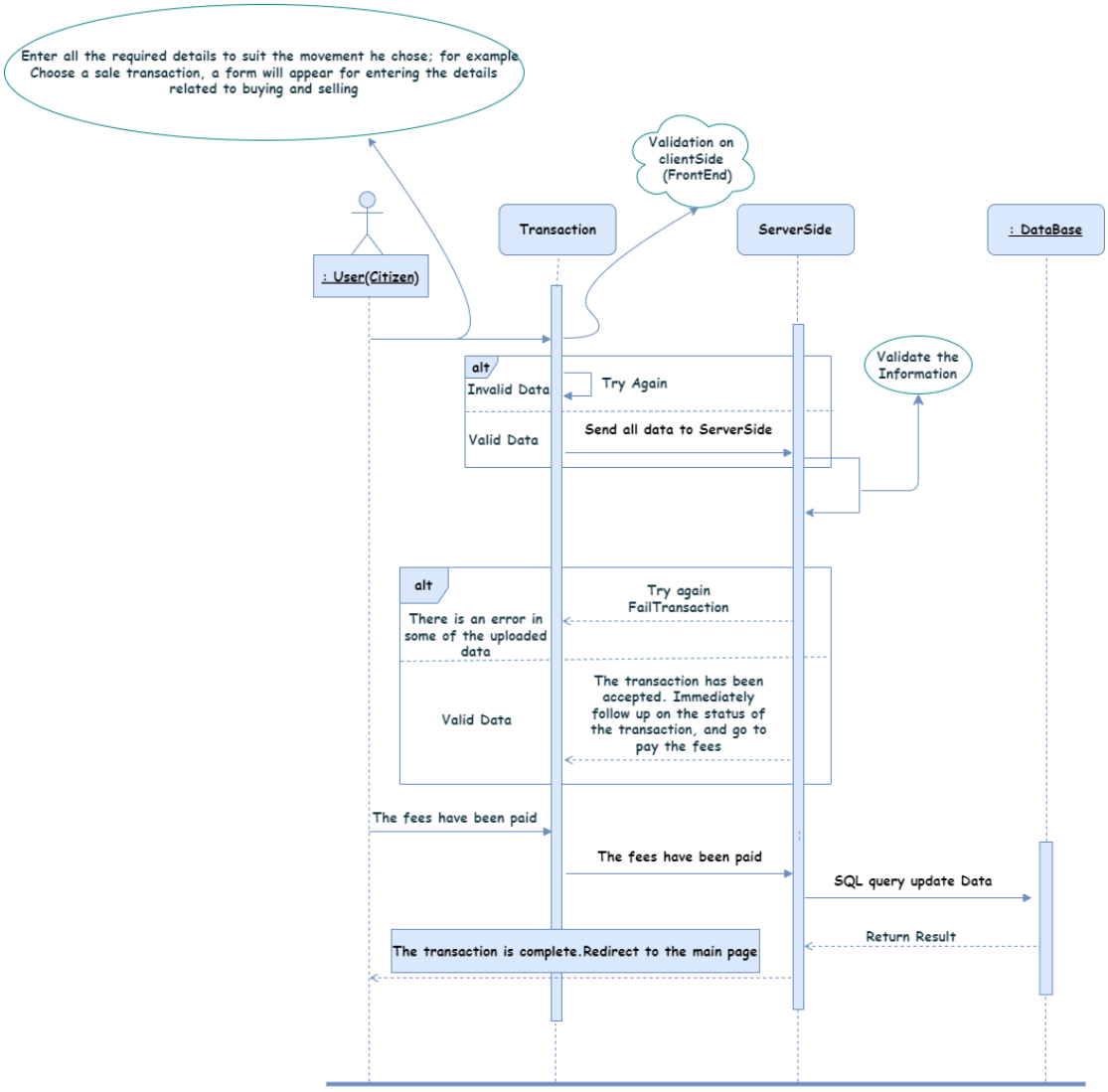


Figure 10: User System Sequence Diagram

6.3 Log-Out

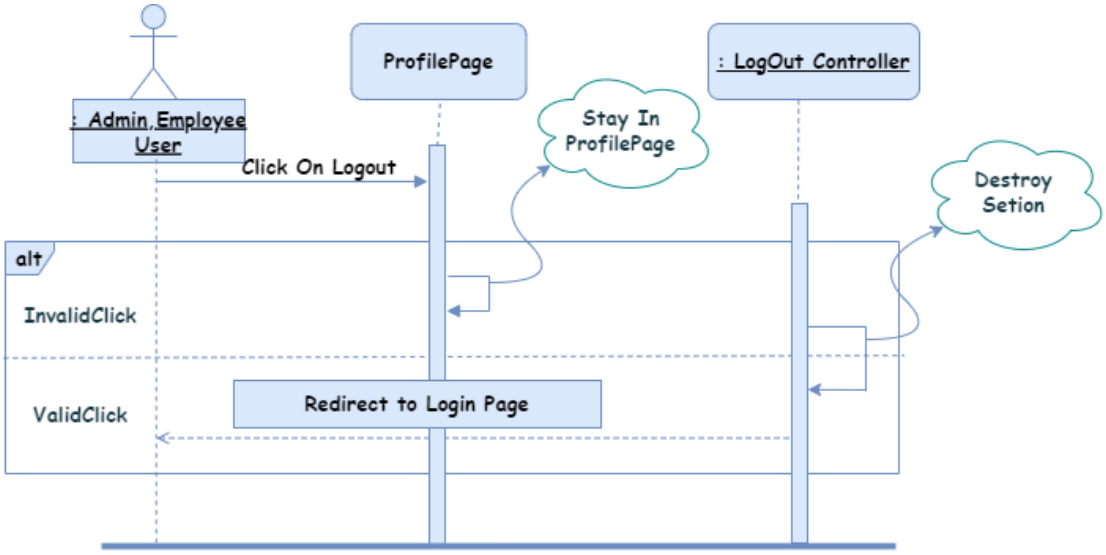
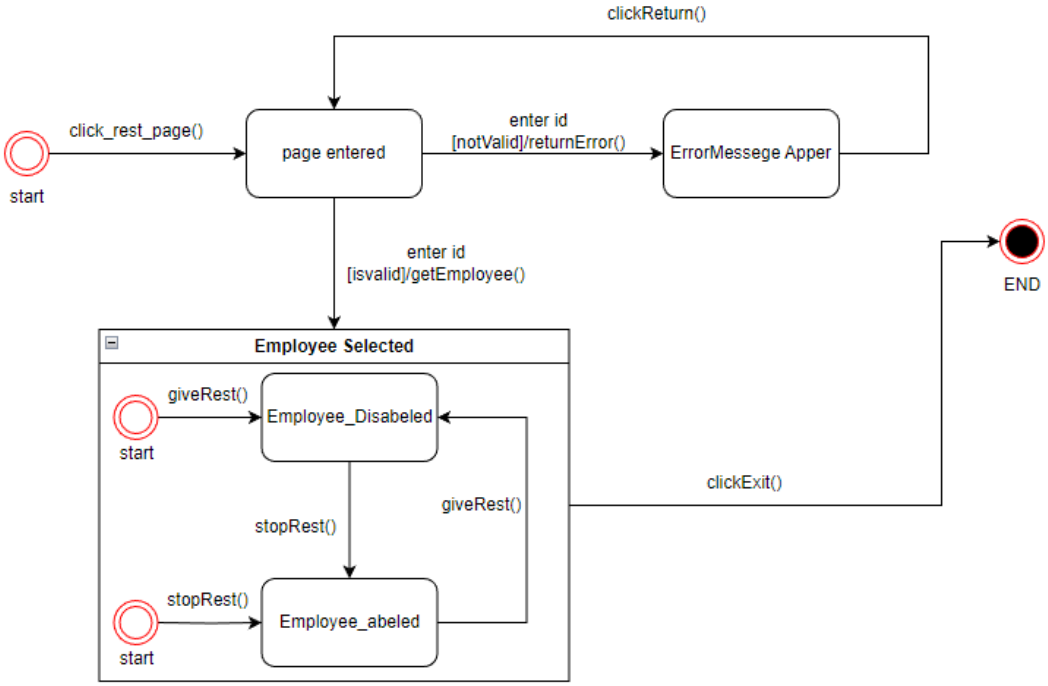


Figure 11: LogOut System Sequence

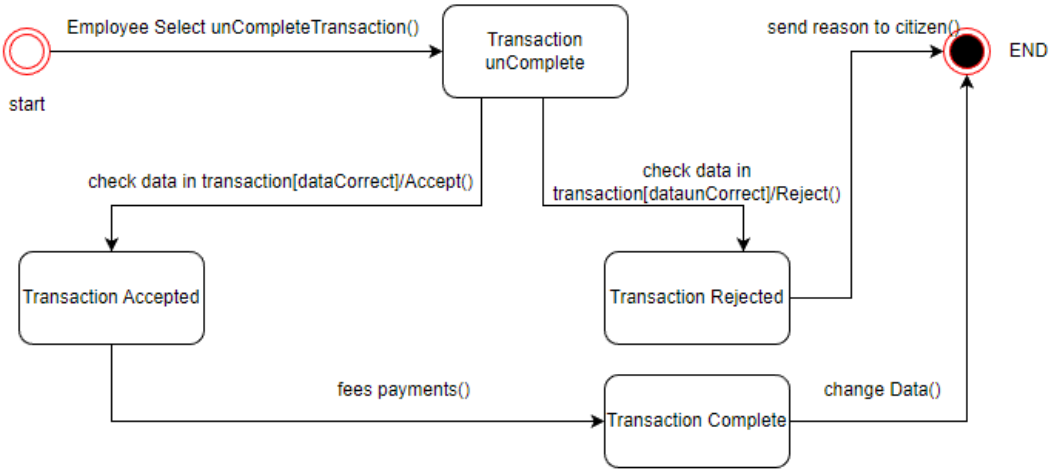
7. State Machine Diagram

7.1 Log-In-to-system



8. *Figure 12: Employee Log-In State Machine*

7.2 Check Transaction



8 *Figure 13: check TX State Machine*

7.3 Send Transaction

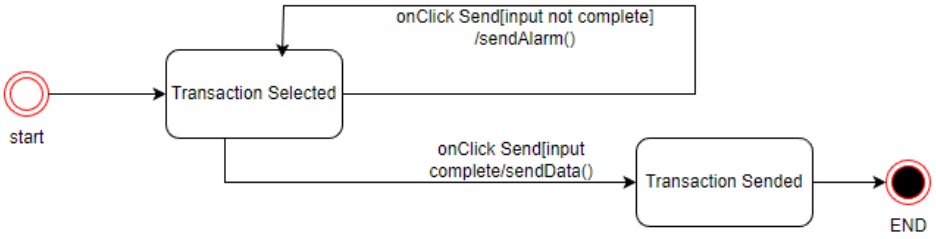


 Figure 14: Send TX State Machine

8. Network Diagram

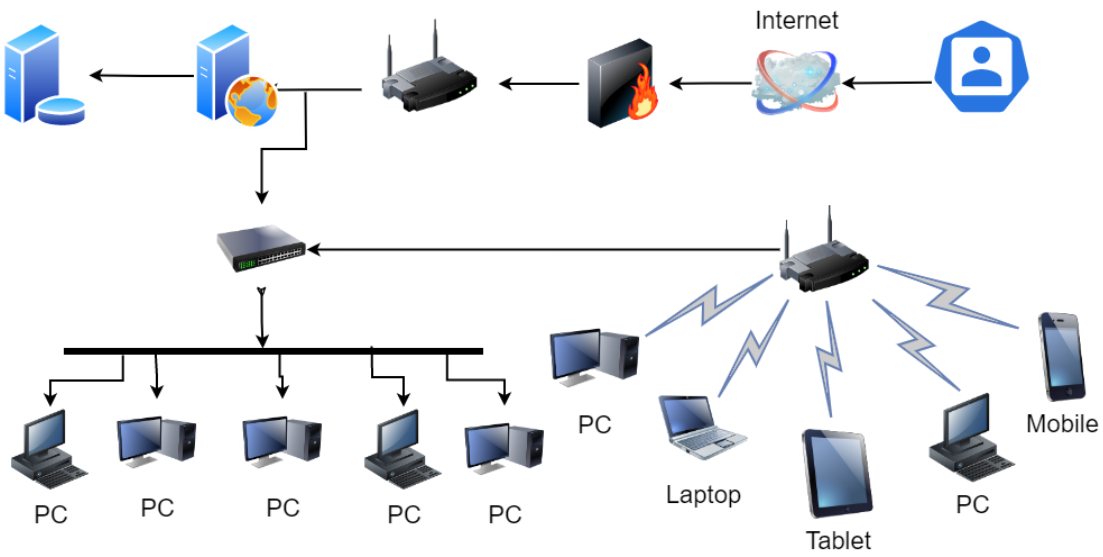


Figure 12: Network Dagram

9. Dialog design and GUI interface.

9.1 Log-In

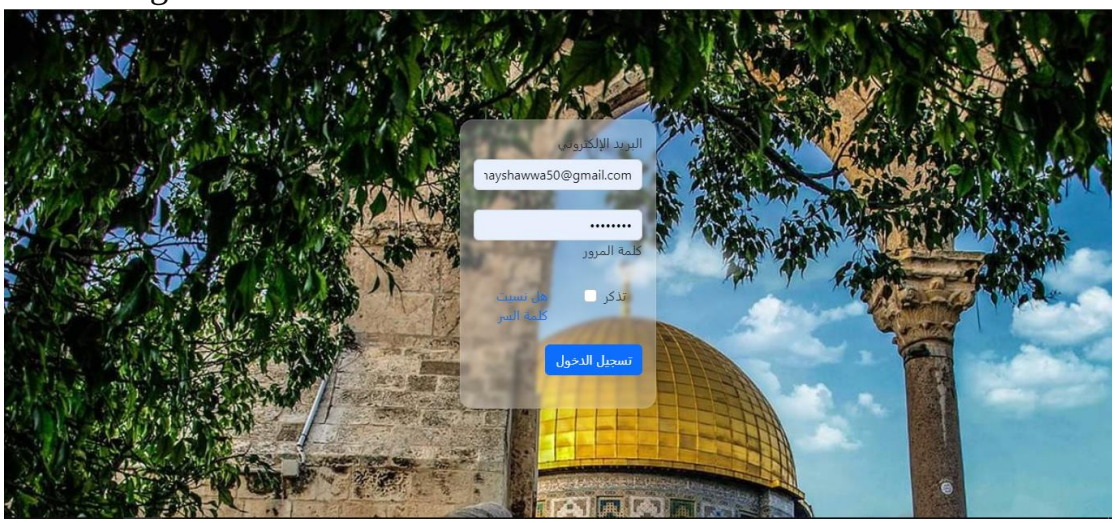


Figure 13: LogIn GUI

System: what would you like to do?

User: I'd like to Log in to my account

system: Ok, I will show you a form, and you have to enter your email and password

User: ok, here is "mohammadenierat@gmail", password is "mohammad21M#e"

System: Unfortunately, the email is invalid. Check it, and try again.

User: Fine, "mohammadenierat@gmail.com", password is "mohammad21M#e"

System: You clicked *LogIn*.

User: yes

system: You are now at your account. What's next?

9.2 Create Employee Account

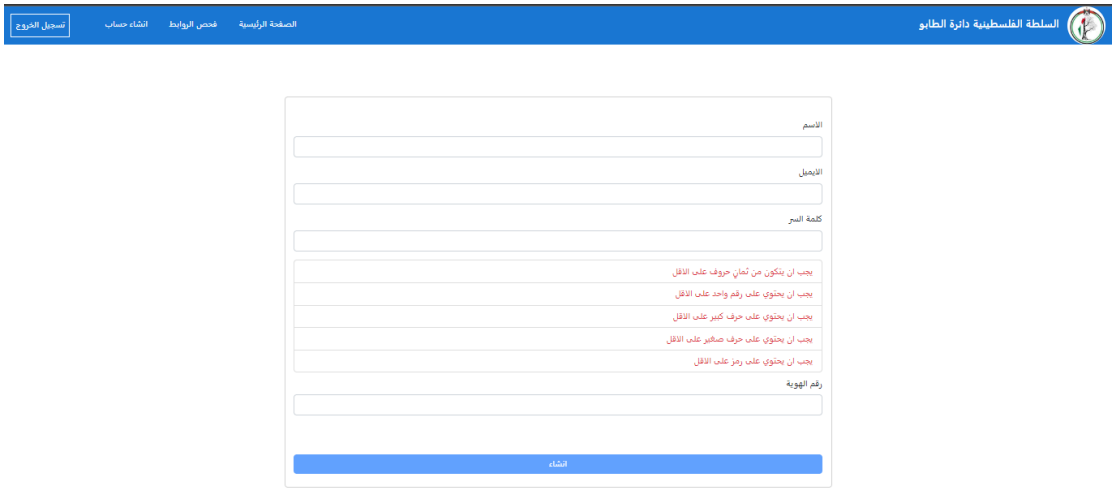


Figure 14: Create Employee GUI

System: what would you like to do?

Admin: I'd like to add (register) a new employee.

System: Here is the registration form, kindly fill out the employee's basic info: user name, email, password, ID number.

Admin: Done

System: The email "mayshawwa50@gmail.com" is taken, please change it.

Admin: Ok, what about " mayshawwa@gmail.com"?

System: The email is valid, but the password must contain at least eight characters, one number, an uppercase letter, a lowercase letter, and a symbol

Admin: *changes the password*

Admin: *Clicked *Create* button*

System: The new employee has been successfully registered. An activation email sent to: mayshawwa@gmail.com, including the username and randomly generated password to login.

9.3 Edit user Account by Employee

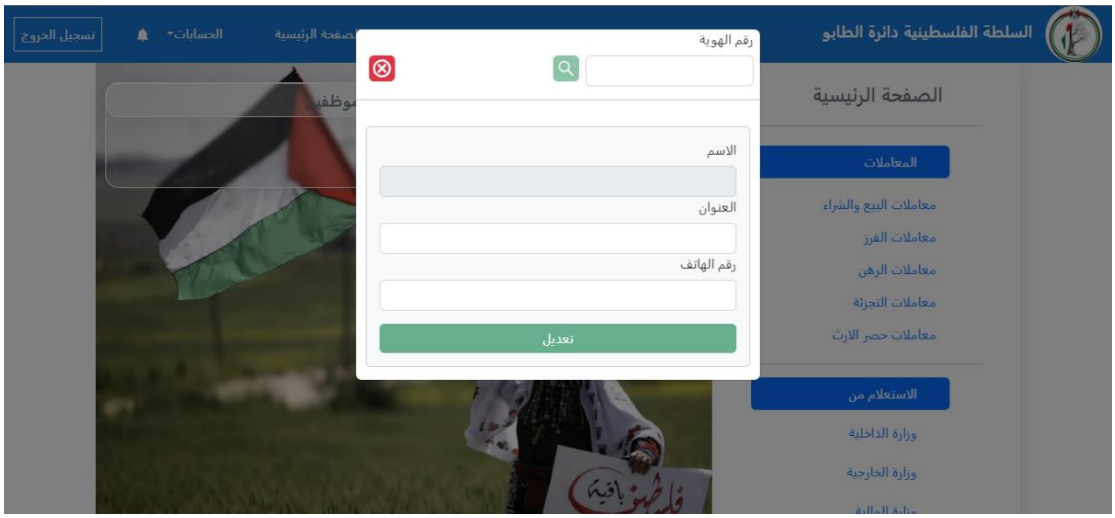


Figure 15: Edite Account GUI

System: what would you like to do?

User: I'd like to update the data for a specific user

system: Ok, I will show you a form, and you have to search for the intended user.

User: I will search by ID

System: Kindly type the ID here in this field.

User: The ID is: "1234567891"

System: Fine, here is "1234567891's info. Click on the field you want to update to enable update mode, write a new value, then click save.

User: I am done. Save.

System: You clicked *Save.* Are you sure you want to save changes?

User: yes

system: Changes have been saved. Do you want to search again for another user?

User: no

10. Design Class Diagram

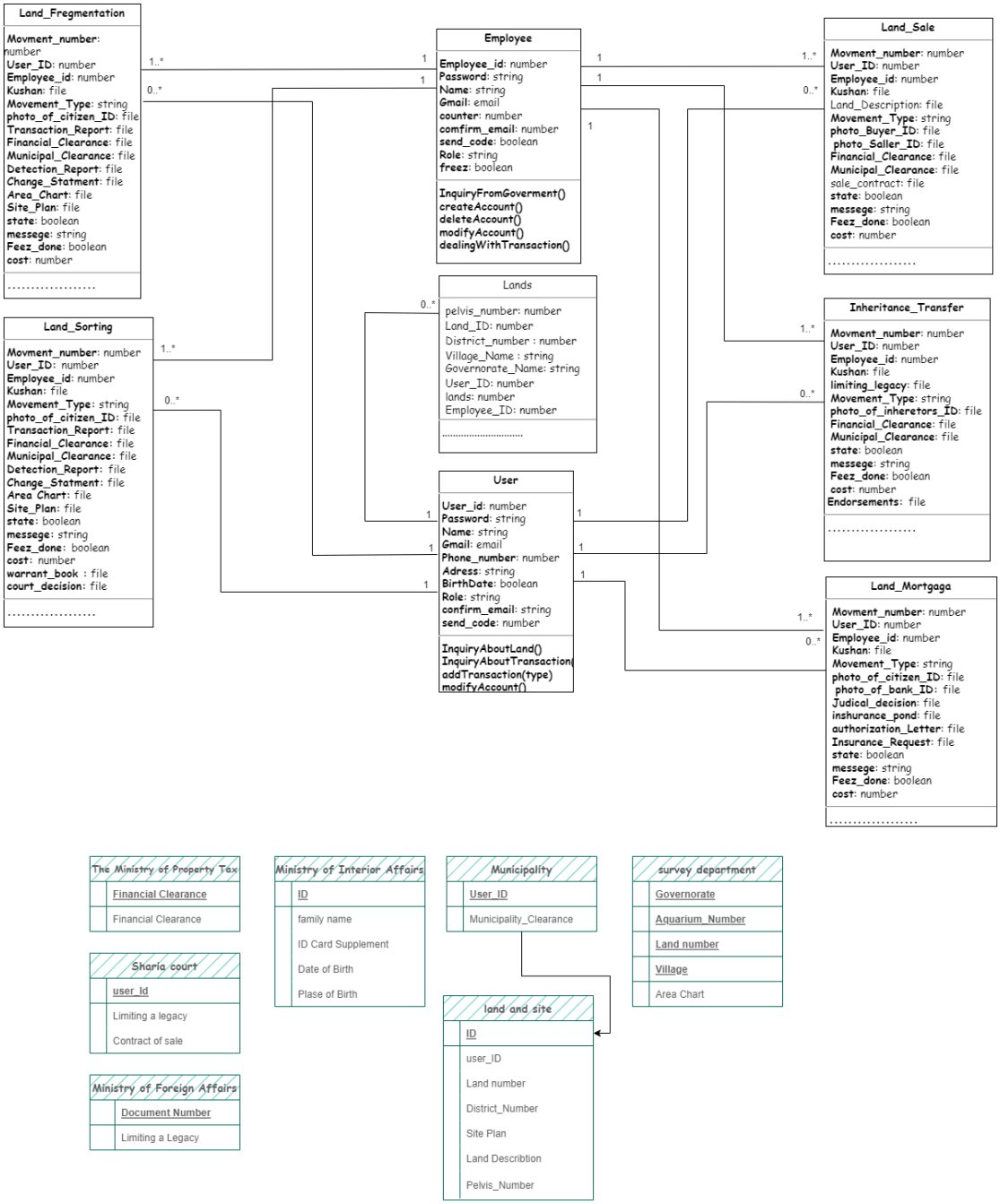


Figure 16: Class Diagram

11. Explanation of the Deployment Plan

1-Since our web application is a three-tier app, typical servers to deploy a three-tier application are web server, application server, and database server, each serves appropriate functional requirements. Each tier operates

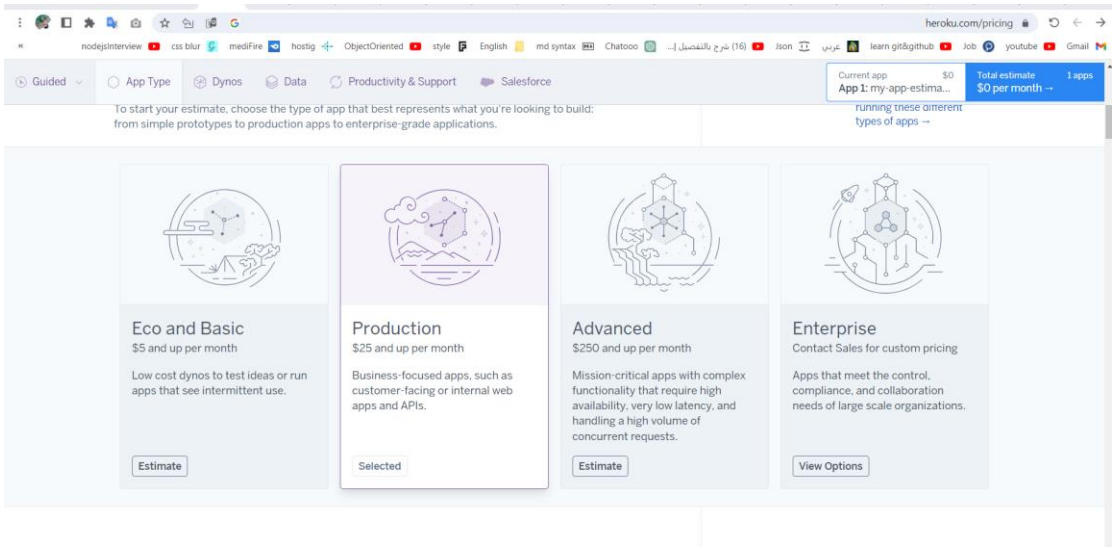
separately, so its services can be customized and optimized without affecting the other tiers.

- ✓ **Speed up the development:** As each tier can be enhanced simultaneously by different teams, so time to market of the product is optimized, and developers can use the latest tools and the best languages for each tier.
- ✓ **Deployment Tasks:** The plan breaks down the deployment process into specific tasks or activities. Each task should be well-defined and include a clear description of what needs to be done, who is responsible, and any associated dependencies or prerequisites. Examples of deployment tasks include installing hardware or software components, configuring databases, setting up network infrastructure, and integrating with existing systems.
- ✓ **Deployment Schedule:** A timeline or schedule is created to outline when each deployment task will occur. It helps in coordinating resources, managing dependencies, and ensuring that the deployment process stays on track. The schedule should consider any critical milestones or deadlines that need to be met.
- ✓ **Improve the scalability:** By deploying the application on different tiers, you are able to scale any tier independently of the others at any given time.
- ✓ **Improve the reliability:** Because it has different tiers, you can also boost reliability and availability by running disparate parts of your application on distinct servers and employing cached results.
- ✓ **Improve the security:** By using a well-designed application tier, it functions as a sort of internal firewall, which will help to prevent SQL injections and other malicious exploits.
- ✓ **Training and Communication:** If necessary, the plan includes provisions for training end-users or administrators on the new system. It outlines the training approach, materials, and schedule. Additionally, it defines the communication channels and methods to be used for informing stakeholders about the deployment progress, changes, and any necessary support.
- ✓ **Testing and Acceptance:** The plan describes the testing and acceptance criteria that need to be met before the system can be considered fully deployed. It may include user acceptance testing, performance testing, security testing, and any other relevant quality assurance measures.
- ✓ **Testing and Acceptance:** The plan describes the testing and acceptance criteria that need to be met before the system can be considered fully deployed. It may include user acceptance testing, performance testing, security testing, and any other relevant quality assurance measures.

2- Our web application was created as a graduation project, and we intend to publish it as a real land management system in Palestine starting from Tulkarm. We just want to deploy this web application to launch it, test it in a production environment, and in order to ease the citizens and employees in the government, so the system needs to start as a trial before we launch it. /shared server would be a good choice as heroku.

Here you can see the paid plans for this server

https://www.heroku.com/pricing


heroku.com/pricing
Guided
App Type
Dynos
Data
Productivity & Support
Salesforce
Current app $0
App 1: my-app-estima...
Total estimate 1 apps
$0 per month →
To start your estimate, choose the type of app that best represents what you're looking to build:
from simple prototypes to production apps to enterprise-grade applications.
running these different
types of apps →
Eco and Basic
$5 and up per month
Low cost dynos to test ideas or run apps that see intermittent use.
Estimate
Production
$25 and up per month
Business-focused apps, such as customer-facing or internal web apps and APIs.
Selected
Advanced
$250 and up per month
Mission-critical apps with complex functionality that require high availability, very low latency, and handling a high volume of concurrent requests.
Estimate
Enterprise
Contact Sales for custom pricing
Apps that meet the control, compliance, and collaboration needs of large scale organizations.
View Options