# Chapter 4

# System Design

**Chapter 4:** System Design

In this chapter we have attached some UML diagrams like Architecture, Domain Model, Class Diagram, Activity Diagram, Component Diagram, Sequence Diagram, Deployment Diagram, ER Diagram etc. By the help of these diagrams we have explained about our whole system that how it works. The each and every function of our system has been explained in this chapter by the help of these diagrams.

## Architecture Diagram

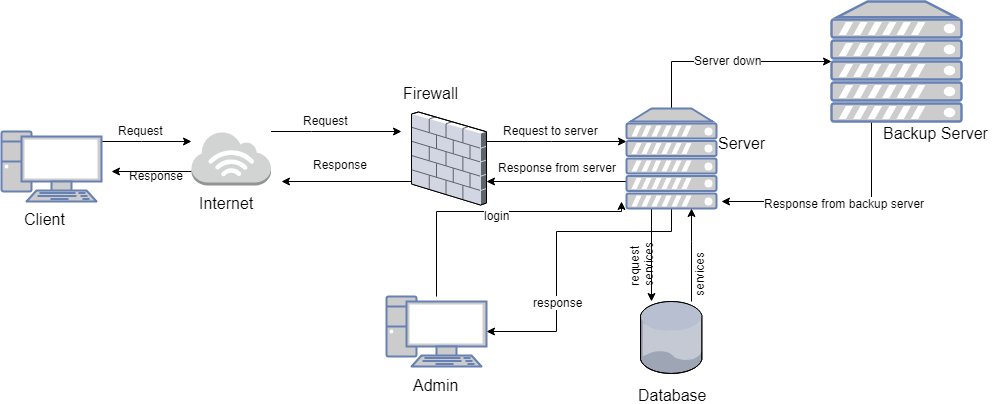


Figure 3 Architecture Diagram

## Domain Model

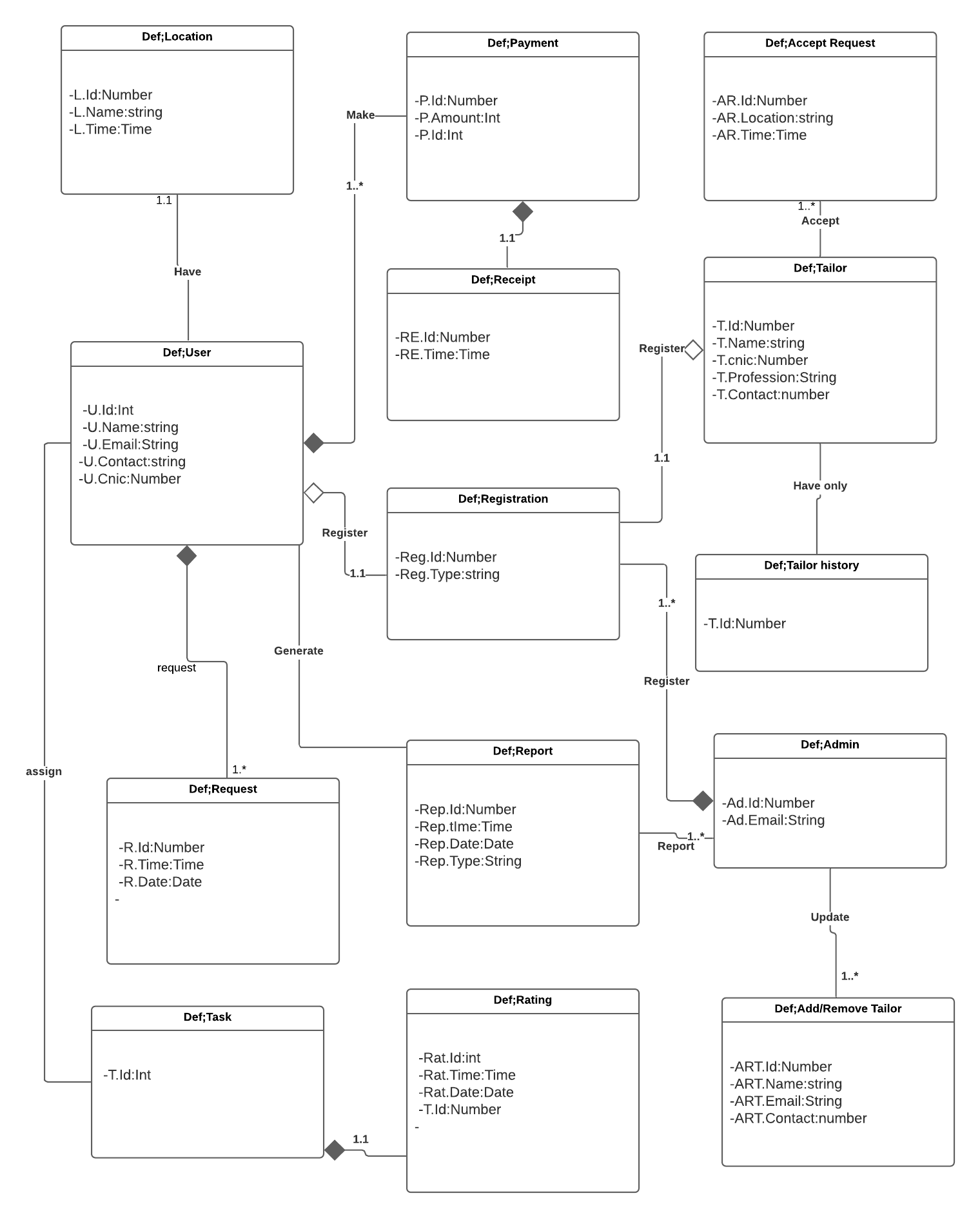


Figure 4 Domain Model

## Entity Relationship Diagram with data dictionary

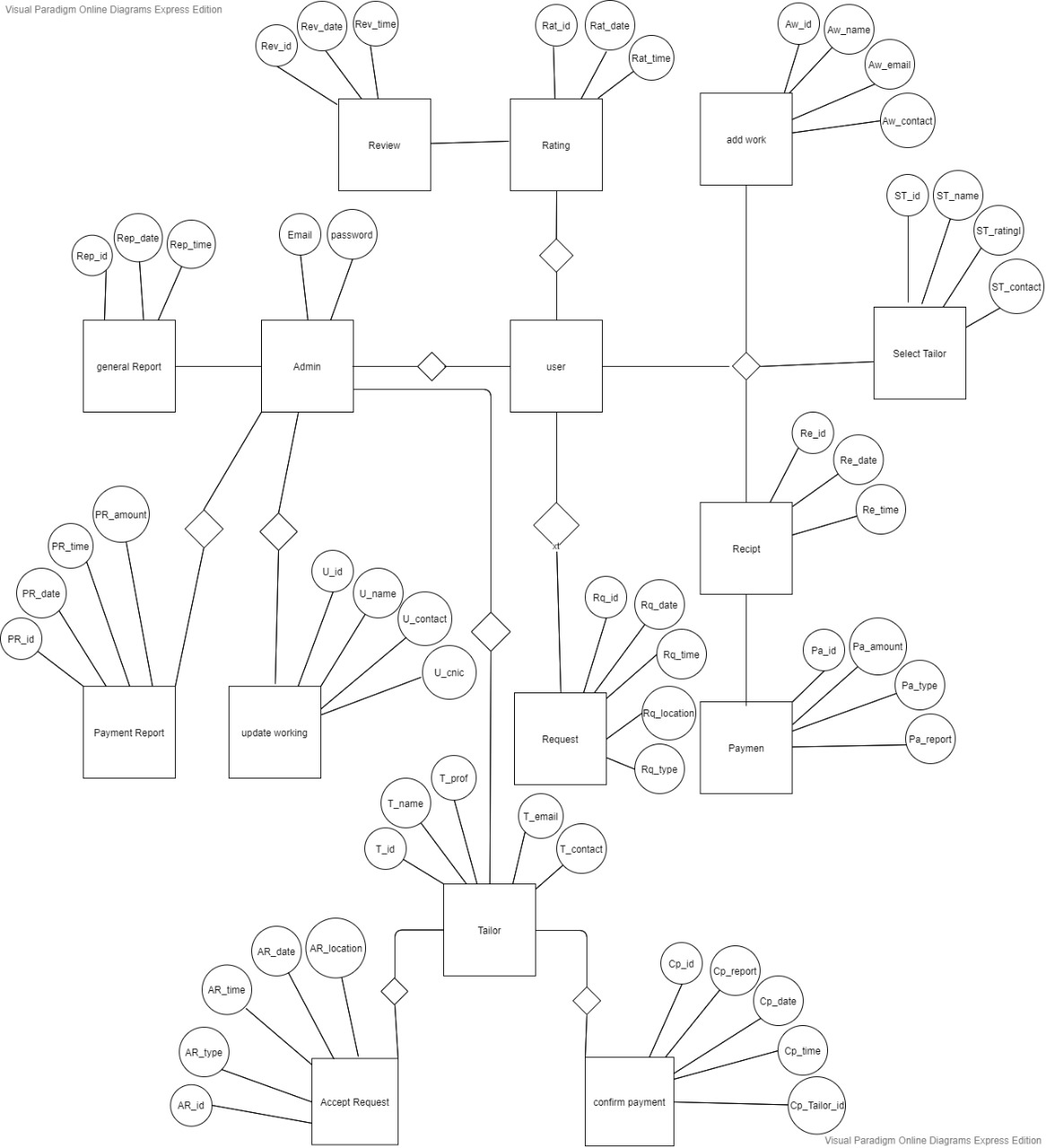


Figure 5 ER Diagram

## Class Diagram

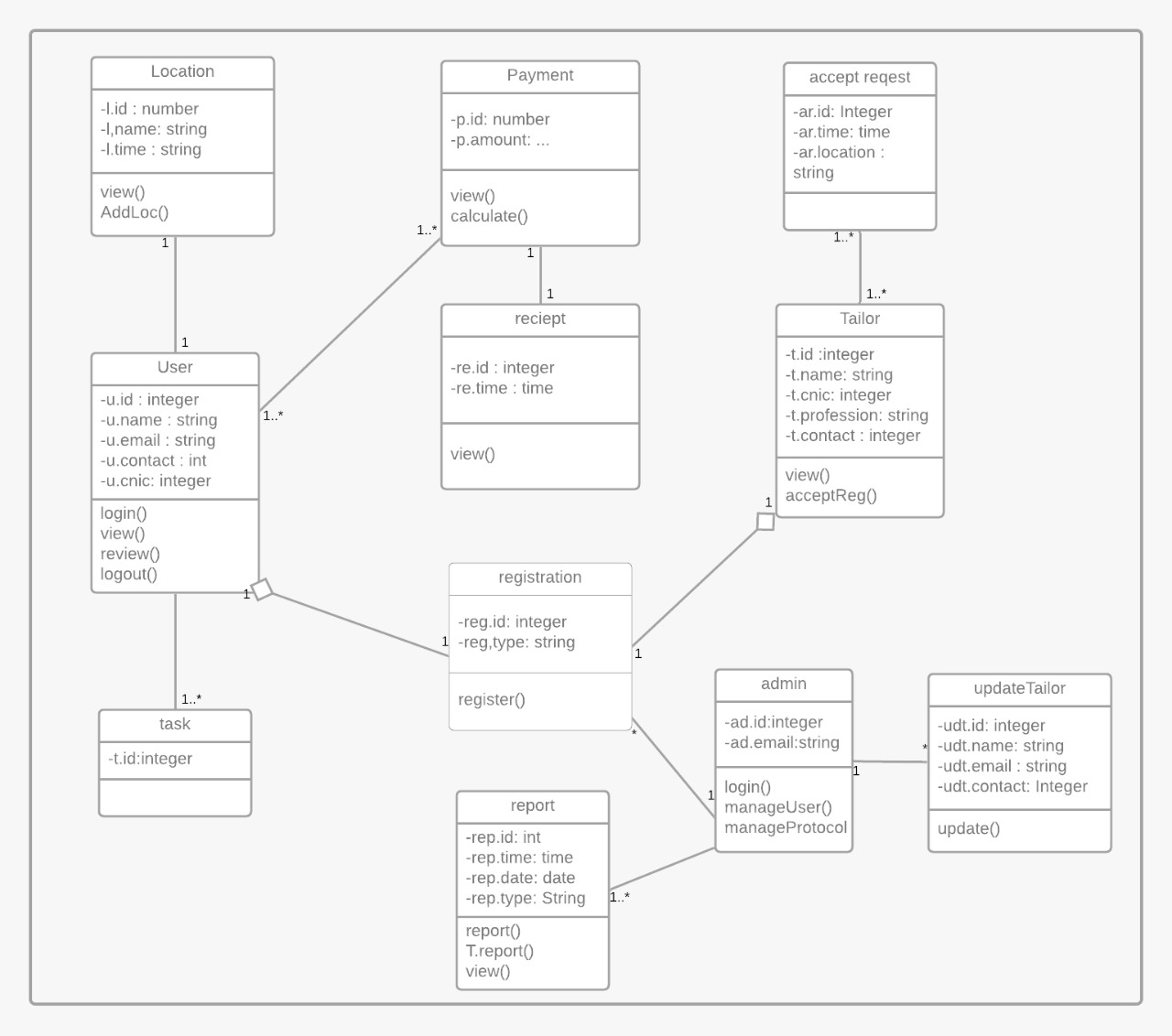


Figure 6 Class Diagram

## Sequence / Collaboration Diagram

1. **Login**

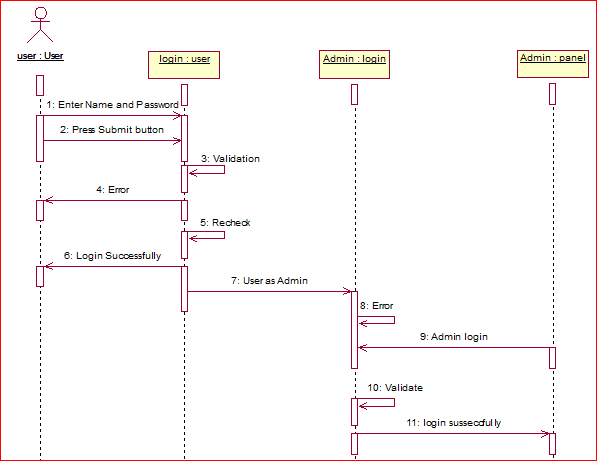
****

Figure 7 Sequence Diagram Login

1. **Registration**

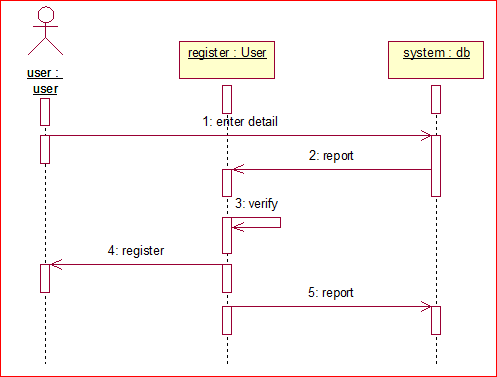
****

Figure 8 Sequence Diagram Registration

1. **Add work**

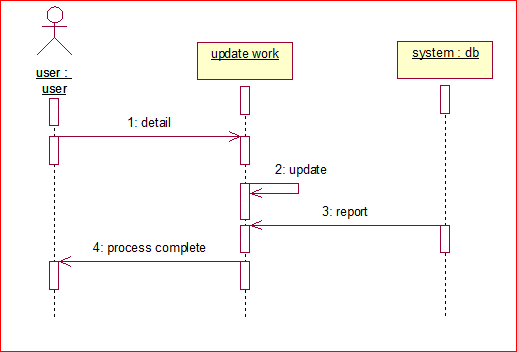
****

Figure 9 Sequence Diagram Add Work

1. **Request / Cancel Request**

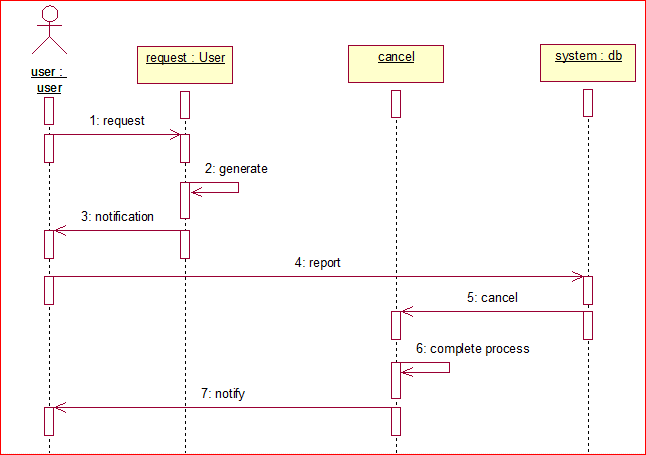
****

Figure 10 Sequence Diagram Request

1. **View profile/ Select Worker**

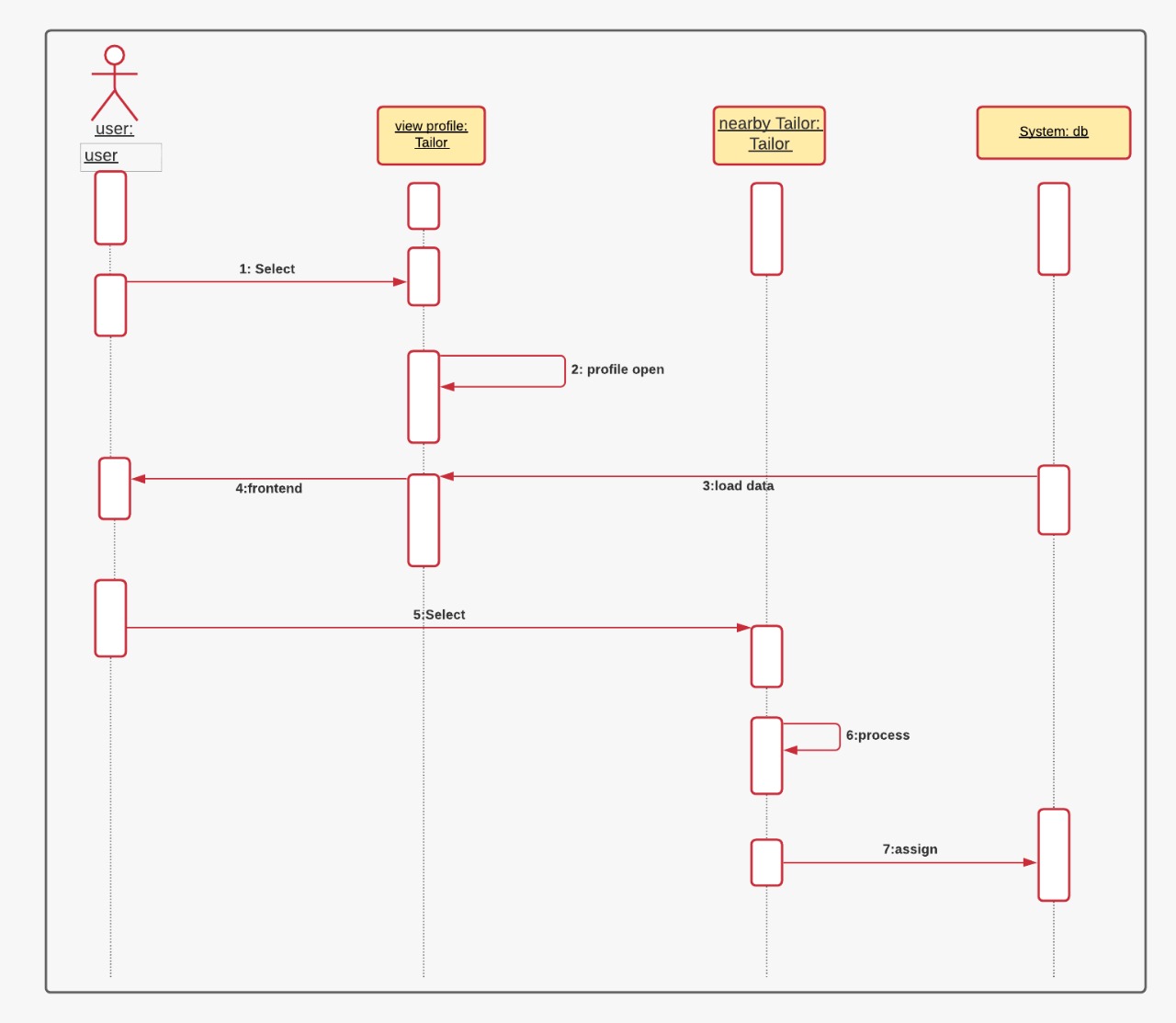
****

Figure 11 Sequence Diagram Profile

1. **Send Notification**

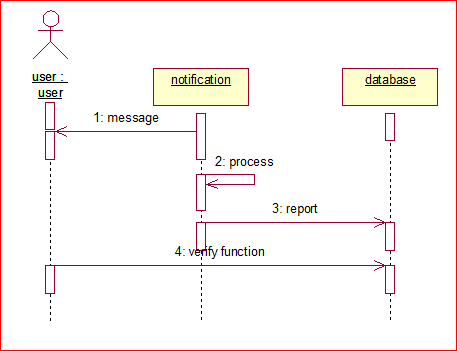
****

Figure 12 Sequence Diagram Notification

1. **Payment**

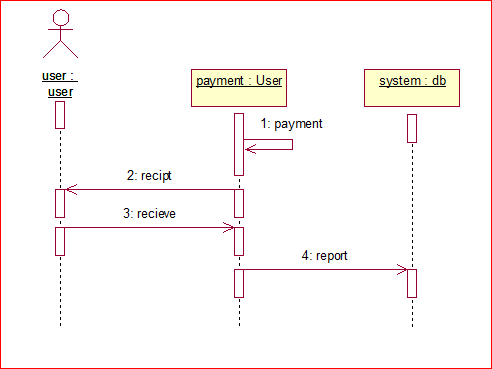
****

Figure 13 Sequence Diagram Payment

1. **Rating/ Review**

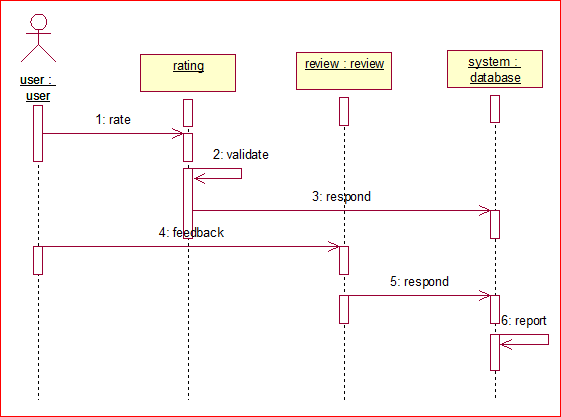
****

Figure 14 Sequence Diagram Rating

1. **Accept Request**

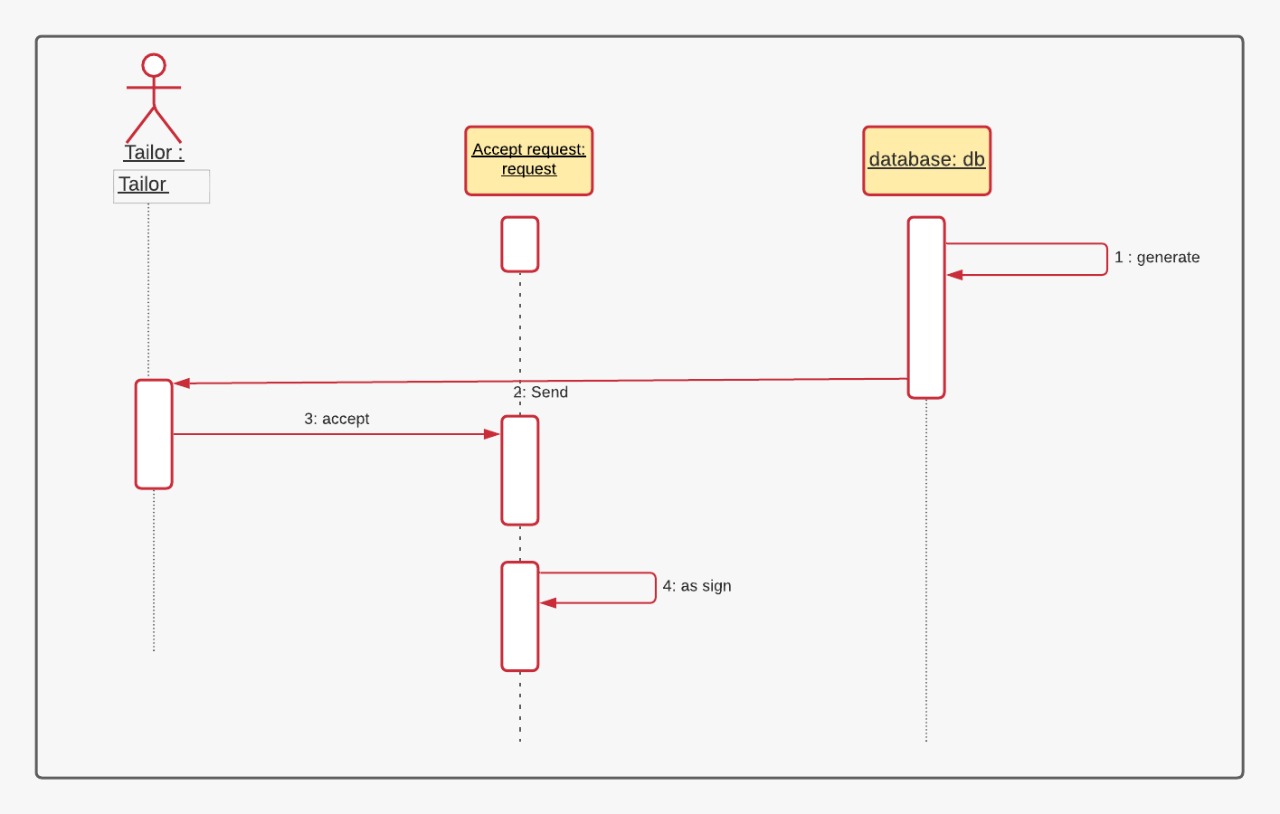
****

Figure 15 Sequence Diagram Accept Request

## Operation contracts

## Operation contracts#1

**Name:** Registration

**Responsibilities:** To register new user.

**Cross References:** Use case(Registration)

**Exceptions:** none

**Preconditions:** Register interface must be open and user must enter valid information.

**Post conditions:** A new registration request instance was created.

* + 1. **Operation contract# 2**

**Name:** Login

**Responsibilities:** User must login.

**Cross References:** Use case(Login)

**Exceptions:** Invalid username and password

**Preconditions:** User must be registered in our system

**Post conditions:**

* A login variable “login” was created
* Instance “login” was associated with current user
* Username and password must match with database of our system
  + 1. **Operation contract# 3**

**Name:** logout

**Responsibilities:** User logout when call logout function

**Cross References:** Use case(Logout)

**Exceptions:** None

**Preconditions:** User must be logged in

**Post conditions:**

* A logout variable “logout” was created
* Variable “logout” was associated with current user
* User successfully logged out
  + 1. **Operation contract# 4**

**Name:** Add work

**Responsibilities:** To add new work or to add a new task

**Cross References:** Use case (Add work)

**Exceptions:** Task didn’t exist

**Preconditions:** User must be logged in

**Post conditions:**

* A variable “add work” was created
* Variable “add work” associated with current user
* Task was successfully added in database
  + 1. **Operation contract# 5**

**Name:** Request

**Responsibilities:** Generate request when user allow

**Cross References:** Use case(Request)

**Exceptions:** System failure

**Preconditions:** User must have data connection and must add a task to perform

**Post conditions:**

* A request variable “request” was created
* Variable request was associated with current user
* Request was successfully added/send to database
* System successfully generate request
  + 1. **Operation contract# 6**

**Name:** View profile

**Responsibilities:** User can view profiles of different Tailor

**Cross References:** Use case (view profile)

**Exceptions:** None

**Preconditions:** User must select registered Tailor

**Post conditions:**

* A variable “view profile” was created
* Variable “view profile” was associated with current user and also with admin
* Profile viewed to user
  + 1. **Operation contract# 7**

**Name:** Select Tailor

**Responsibilities:** User can select Tailor

**Cross References:** Use case (Select Tailor)

**Exceptions:** user select or de-select Tailor

**Preconditions:** User must assign a task to select registered Tailor

**Post conditions:**

* A variable “Tailor” was created
* Variable “Tailor” was associated with current Tailor and also with admin
* A Tailor has been successfully selected
  + 1. **Operation contract# 8**

**Name:** Send notification

**Responsibilities:** Users able to send notification to system and system also respond through notification

**Cross References:** Use case (send notification)

**Exceptions:** Particular user not found/ System failed to receive your message

**Preconditions:** User must be logged in

**Post conditions:**

* A notification variable “notification” was created
* Variable “notification” was associated with user and system
* Notification send successfully
  + 1. **Operation contract# 9**

**Name:** Payment

**Responsibilities:** User give payment to Tailor

**Cross References:** Use case (payment)

**Exceptions:** User pay through credit card to the system

**Preconditions:** Users task must be complete

**Post conditions:**

* A payment variable “payment” created
* Variable payment associated with both Tailor and user
* User successfully paid to Tailor
  + 1. **Operation contract#10**

**Name:** Rating

**Responsibilities:** User can rate to profiles of different Tailor

**Cross References:** Use case (Rating)

**Exceptions:** User not give rating

**Preconditions:** User must select registered Tailor

**Post conditions:**

* A variable “rating” created
* Variable rating was associated with Tailor
* Rating process successfully completed

## Activity Diagram

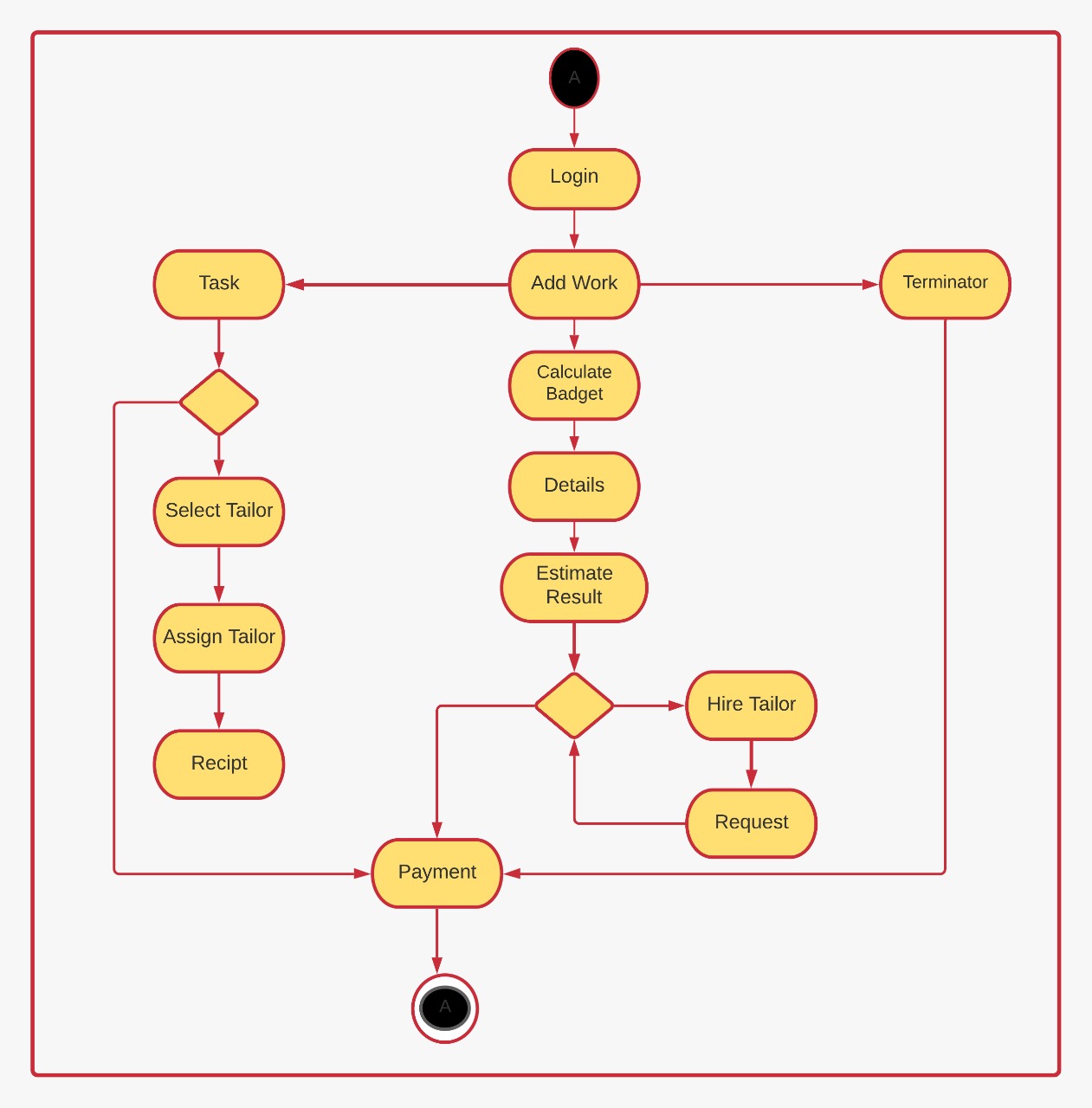


Figure 16 Activity Diagram

## State Transition Diagram

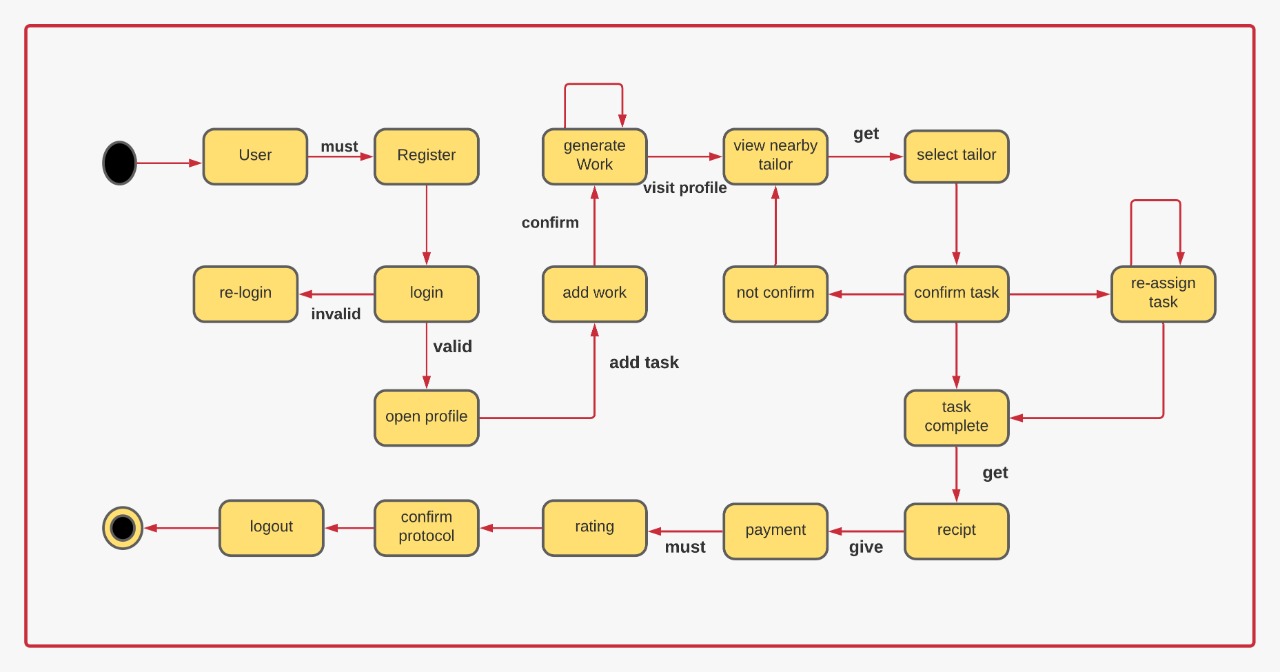


Figure 17 State Diagram

## Component Diagram

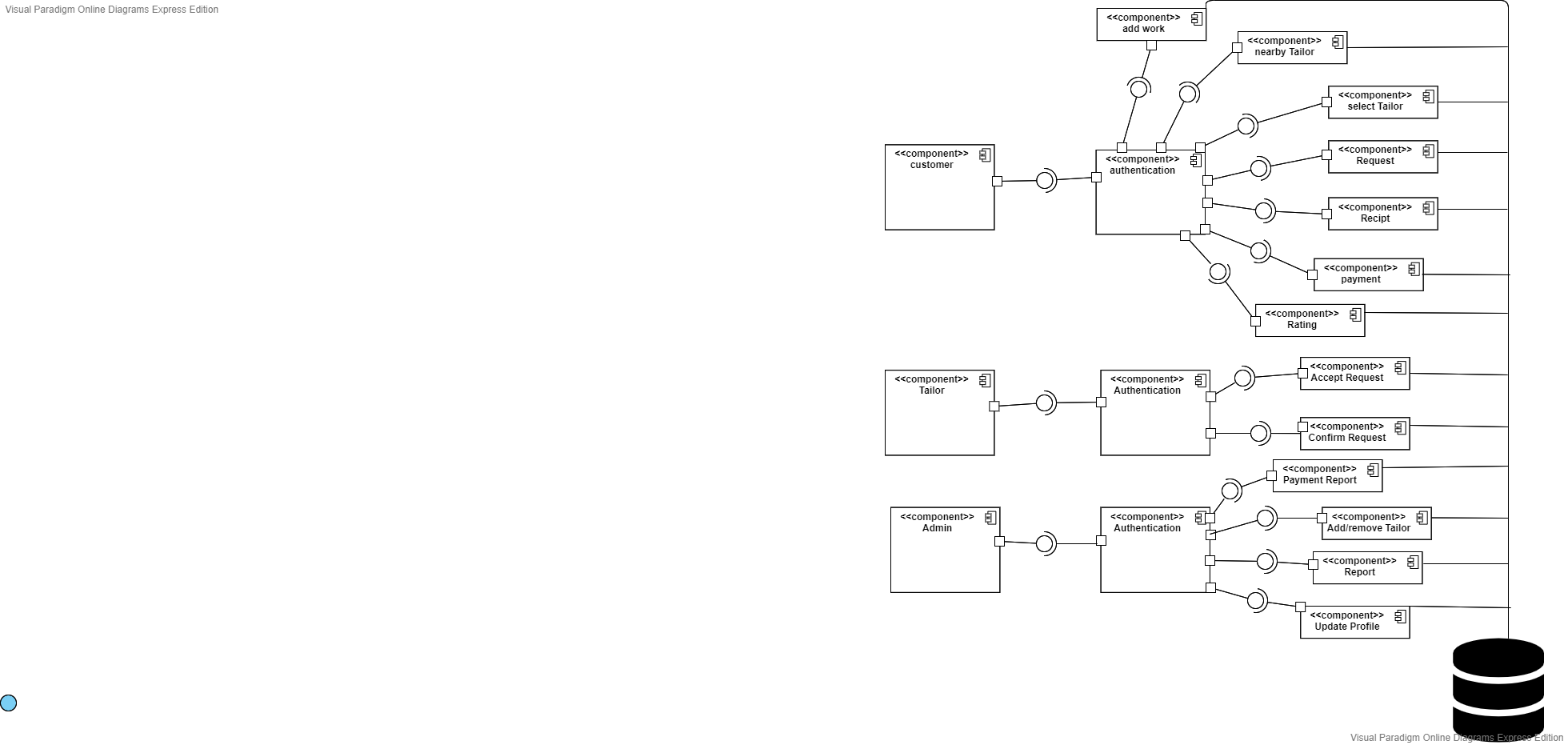
1. 

Figure 18 Component Diagram

## Deployment Diagram

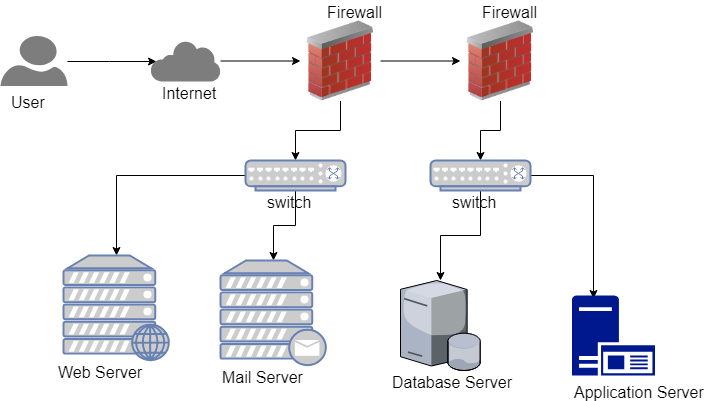


Figure 19 Deployment Diagram

## Data Flow diagram

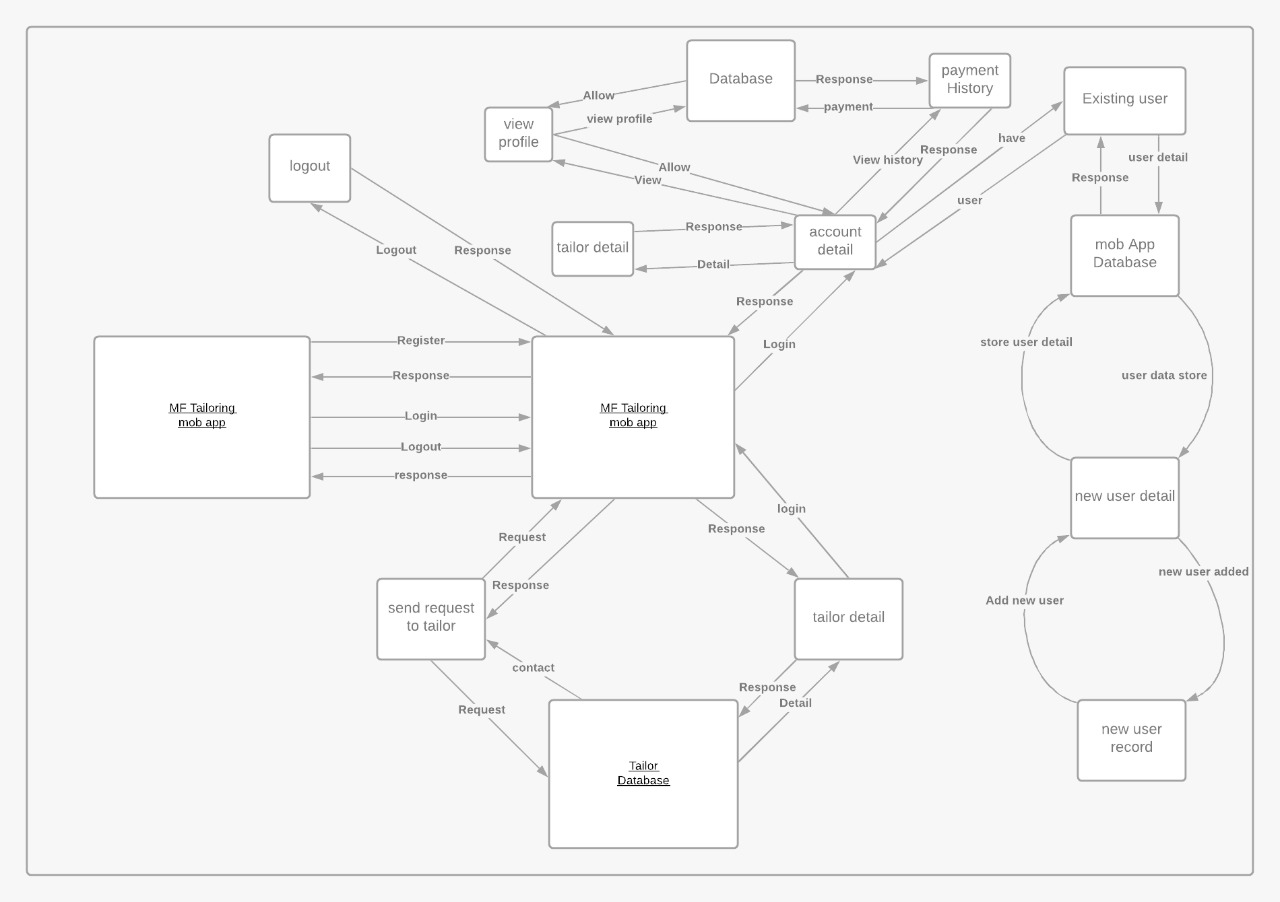


Figure 20 Data Flow Diagram