

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering



Information System Design And Software Engineering Lab,
CSE 3224

Year: 3rd Semester: 2nd

Lab Section: A2

Group: 01

Project Name: **Basha Bari**

Group Members:

Name: Mainul Sajid ID:170204029

Name: Adnan Saky ID:170204035

Name: Julfikar Ibnul Haque ID:170204055

Date of Submission: 2nd February, 2021

Introduction To Project

BashaBari is a website to communicate with house owners and tenants. There are two types of users: one house owner and another tenant. The main goal of the project is to set up a connection between the house owner and tenants via online, where they can share their problems, requests, pay bills and important notices at one place. This platform can help both house owners and tenants from various hassles. So, there is a huge possibility of success for us.

Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional way to visualize the information within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

An Overview of Elements in Data Flow Diagram of Our Project

1. Source/Sink

- House Owner
- Tenants

2. Process

- Registration
- Log in
- Verifying
- Updating Profile
- Posting Notice
- Viewing Notice
- Posting request
- Viewing request
- Posting bill
- Viewing bill
- Paying bill
- Requests owner to add
- Accepts add request

3. Data Store

- User Information
- Transaction information
- Notice
- Requests
- Connected List

Data Flow Diagram

Context level Diagram

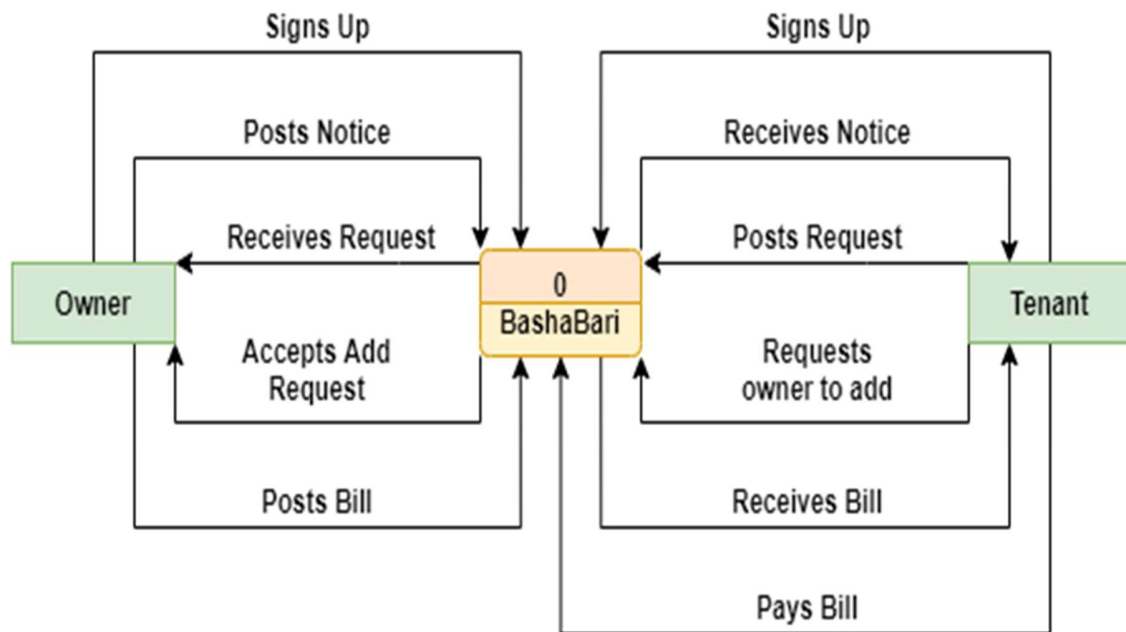


Fig: Context Diagram

Level-0 Data Flow Diagram

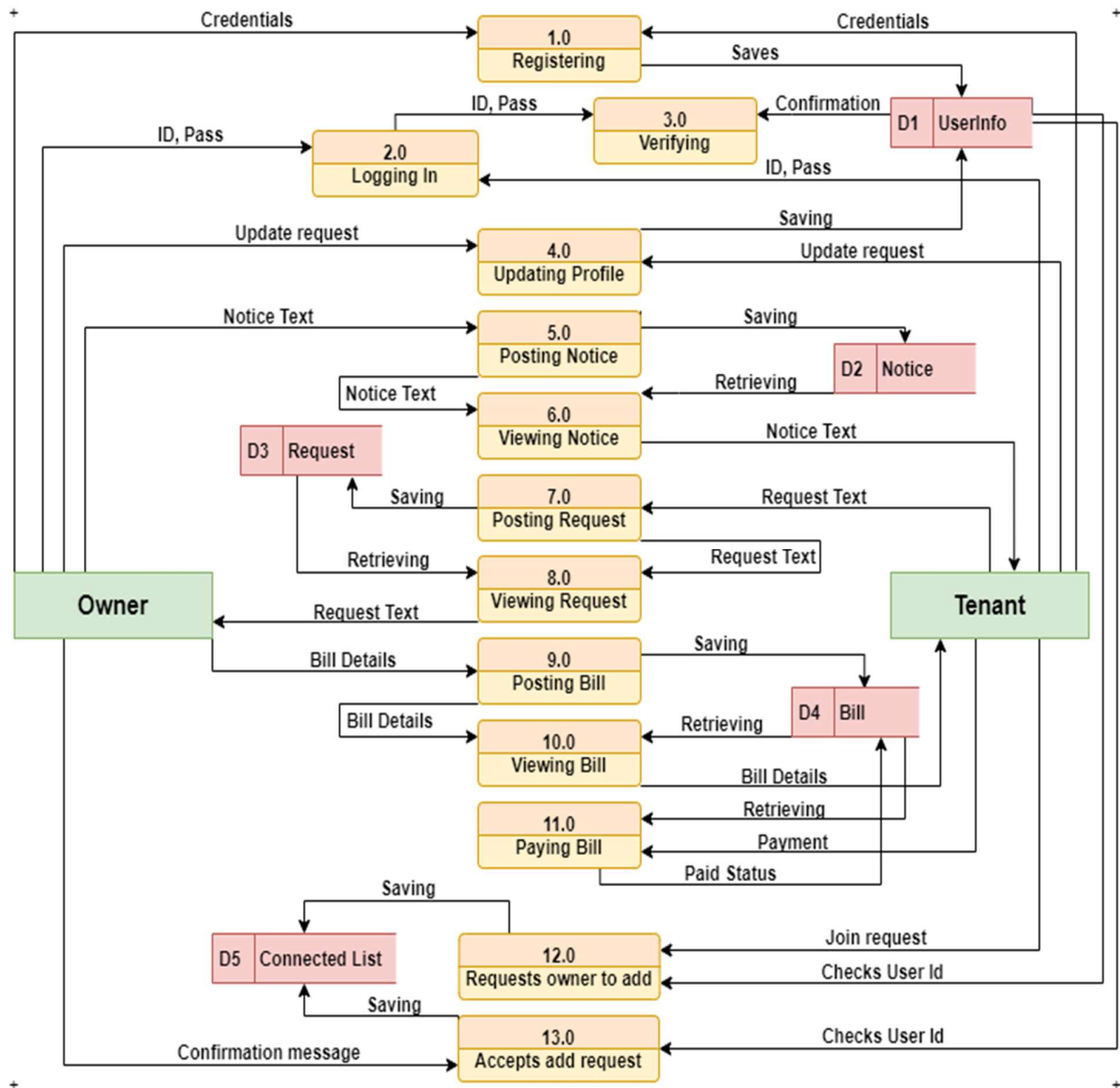


Fig: Level-0 data flow diagram

Class Diagram

UML CLASS DIAGRAM gives an overview of a software system by displaying classes, attributes, operations, and their relationships. This Diagram includes the class name, attributes, and operation in separate designated compartments.

Class Diagram defines the types of objects in the system and the different types of relationships that exist among them. It gives a high-level view of an application. This modeling method can run with almost all Object Oriented Methods. A class can refer to another class. A class can have its objects or may inherit from other classes. Class Diagram helps construct the code for the software application development. The purpose of the class diagram can be summarized as:

- ❖ Analysis and design of the static view of an application.
- ❖ To describe the responsibilities of a system.
- ❖ To provide a base for component and deployment diagrams.
- ❖ Forward and reverse engineering.

Class Diagram of the Project

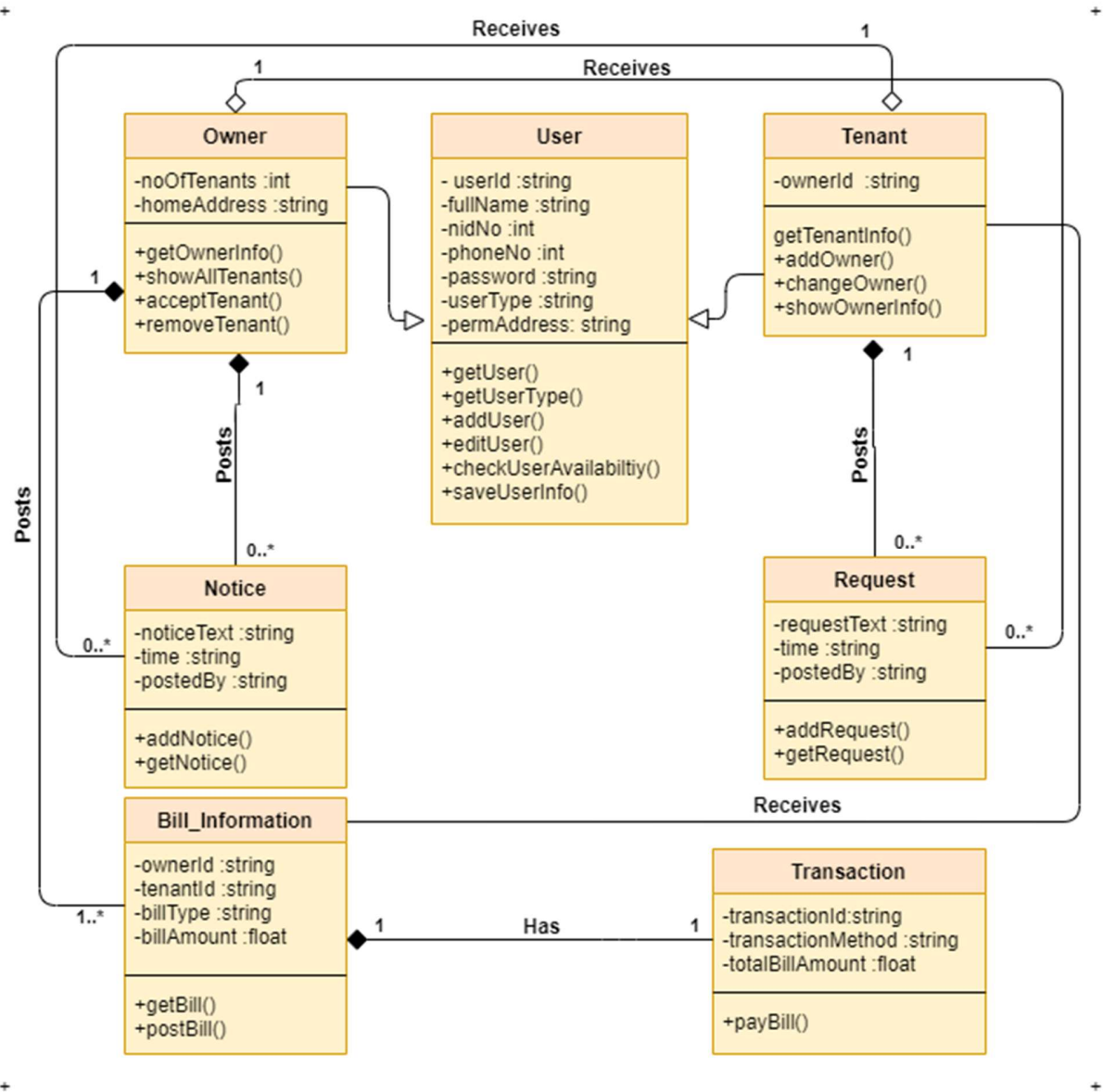


Fig: Class Diagram

Conclusion

Data flow diagrams (DFDs) offer a graphical technique for summarizing the movement of data between the processing steps that occur within a business process. They isolate the collections of data, or data stores, which accumulate during a process, and identify the sources of data that arise outside process boundaries. The Data Flow Diagram will definitely help us to see the inner mechanism of our application and meet the requirements. The Data Flow Diagram will help us to understand what our application needs, what path should we follow, what way our data will be utilized. Data Flow Diagram is an important step of the Software Development Life Cycle. The information we will obtain from here will greatly help us to redesign our application and for that, we have prepared a Data Flow Diagram with Context Diagram and Level-0 Data Flow Diagram to show how our application will function when it is fully completed.

The Class Diagram will help us to analyze the static view of our application as well as design of the static view. It will help to describe the responsibilities of our system. This class diagram will also help us to provide a base for component and also deployment diagrams. Mostly, this diagram will help us to forward and reverse engineering our application.