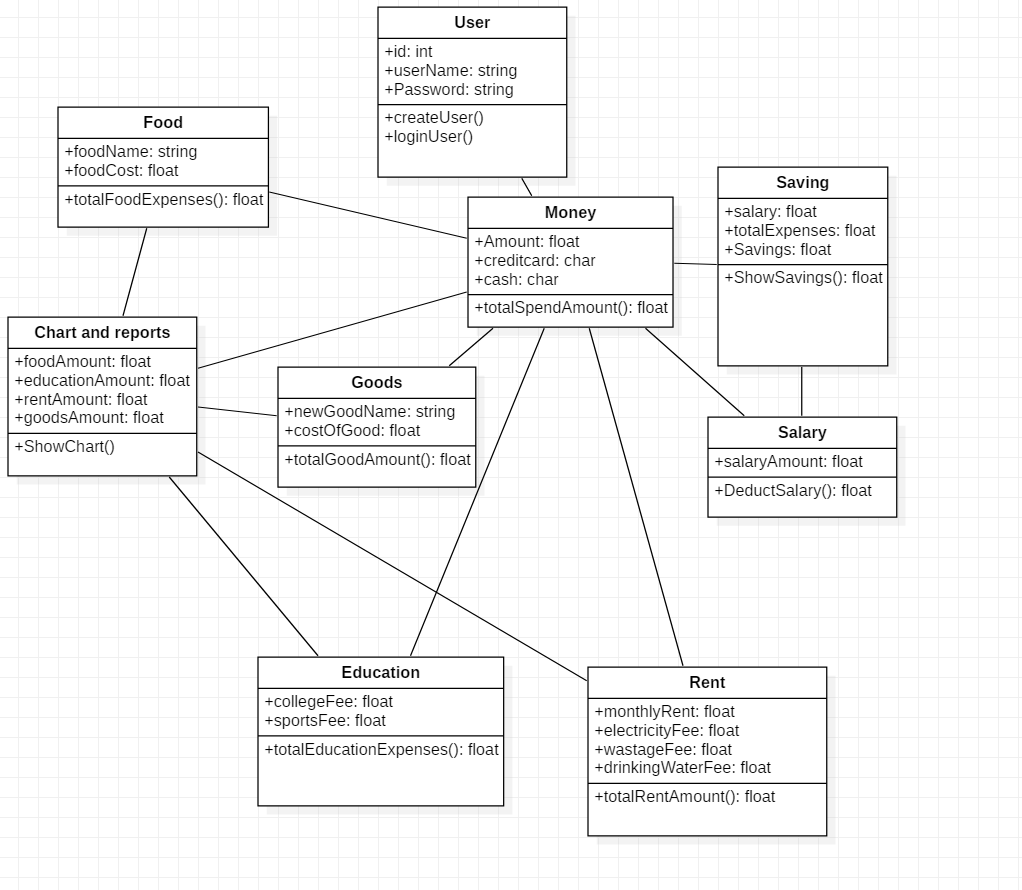
Design

1. Structural design
2. Class diagram

A class diagram is a graphical of representation of the classes that are interrelated with each other. Class diagram helps to show how each class are related with each other. Class diagram will help further in design phases. Class diagram is created in order to clear the doubt during the design process.

Here is the class diagram for this project:



Justification

Class diagram is one of the important architectural diagrams which further helps during the implementation phase of software development. It helps us to define class that can be used as key elements during the software development.

Notation:

Class: There are different class which has its attributes that will help create a SQL database.

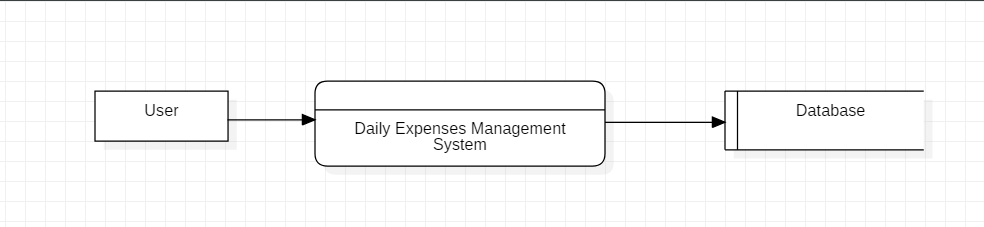
Association:

Association between different class helps to see which classes are interrelated to which class.

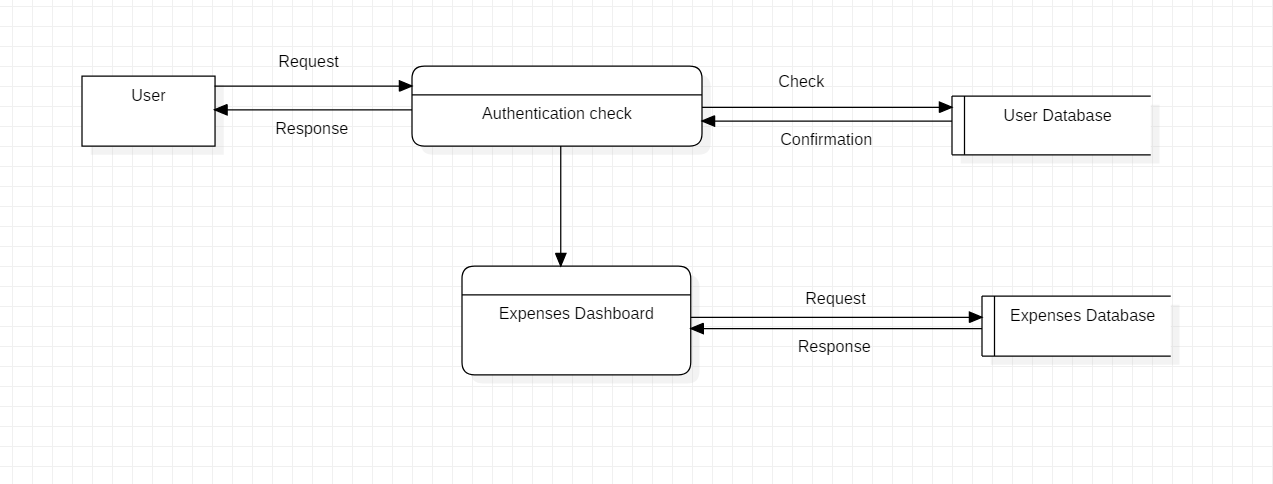
1. Dataflow diagram

Data flow diagram shows how data flows in the system. It is just a graphical representation of how data flows inside the system. Different shapes and notations are used to complete the flow of data. There are different levels of DFD according to their briefness. For this project I m going to create up to level 2 DFD.

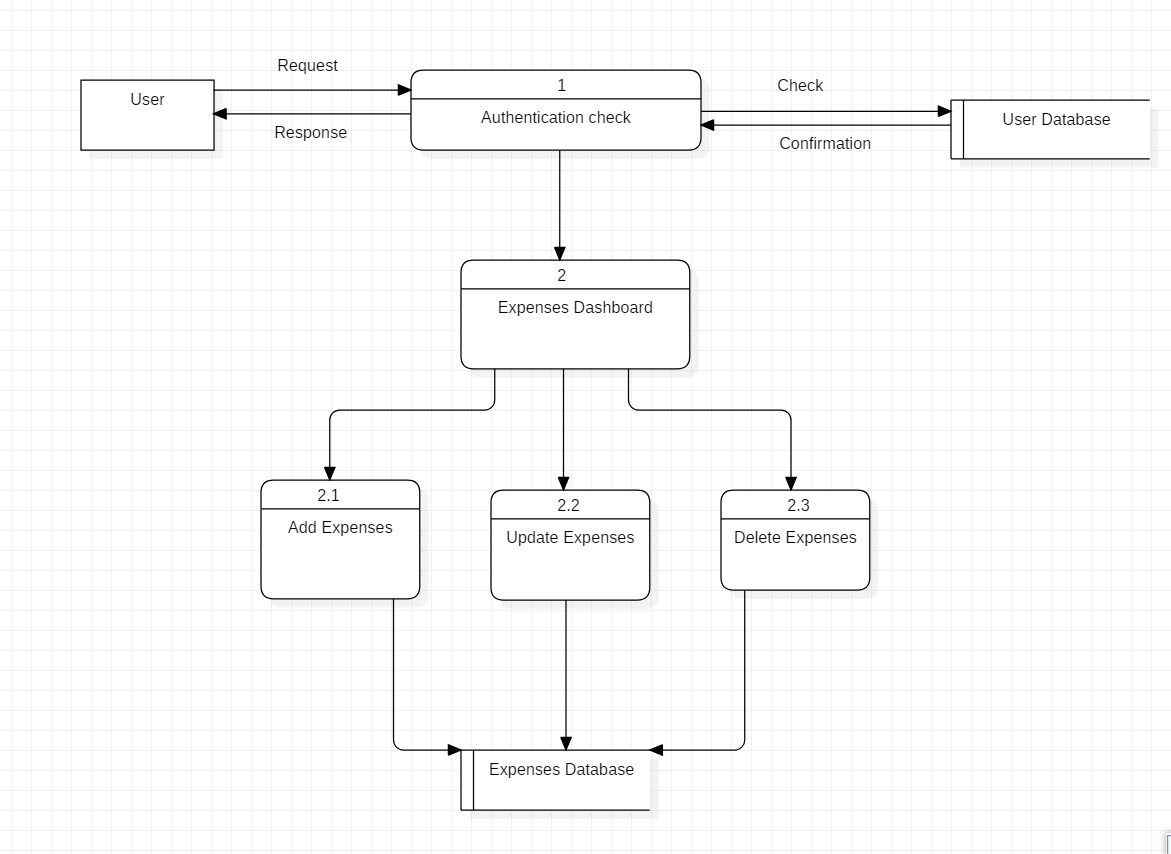
Level 0:



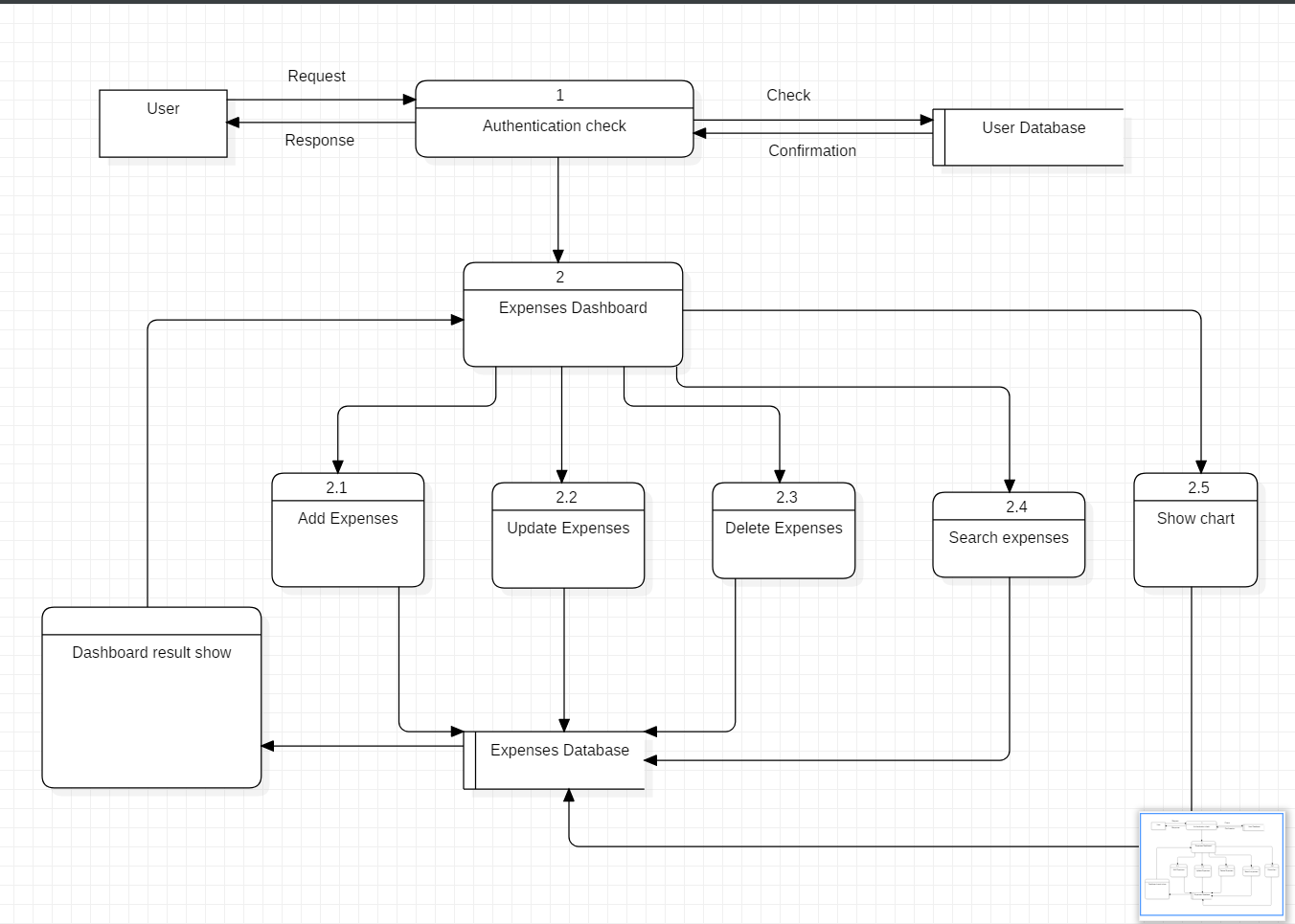
Level 1:



Level 2:



Level 3:



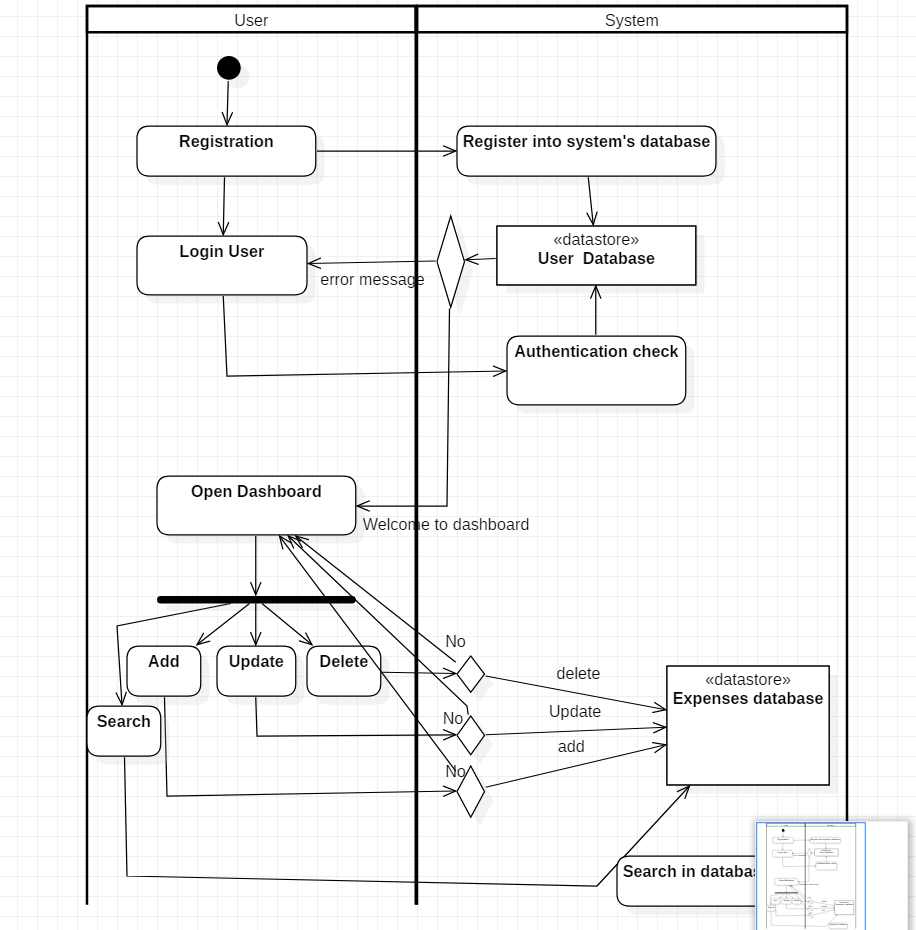
Justification:

Data Flow diagram helps to show how the system data is transferred from one to another. Here above there are different levels of DFD. In level 0 there is user whose data are recorded using the software into its database. It shows only how data is transferred from user to the database. In level 1 DFD the process (authentication check) is further broken down into how user requests his/ her login and how system’s database response to the action. In level2 the dashboard shows what operation can be done by the user after successful login and every change made will be recorded into system’s database. Finally, in level 3 every operation that can be done in the dashboard are shown which finally transfers all the data into the system’ database.

Notation:

1. Entity: entities are used to keep a baseline that will further do some activities.
2. Process: process is notation used to show what activities are happening.
3. Datastore: Datastore is the database where we store data after a process is completed.
4. Flow : Flow is a notation for how a data is flowing from one to another.
5. Behavioral design
6. Activity

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, and so on.



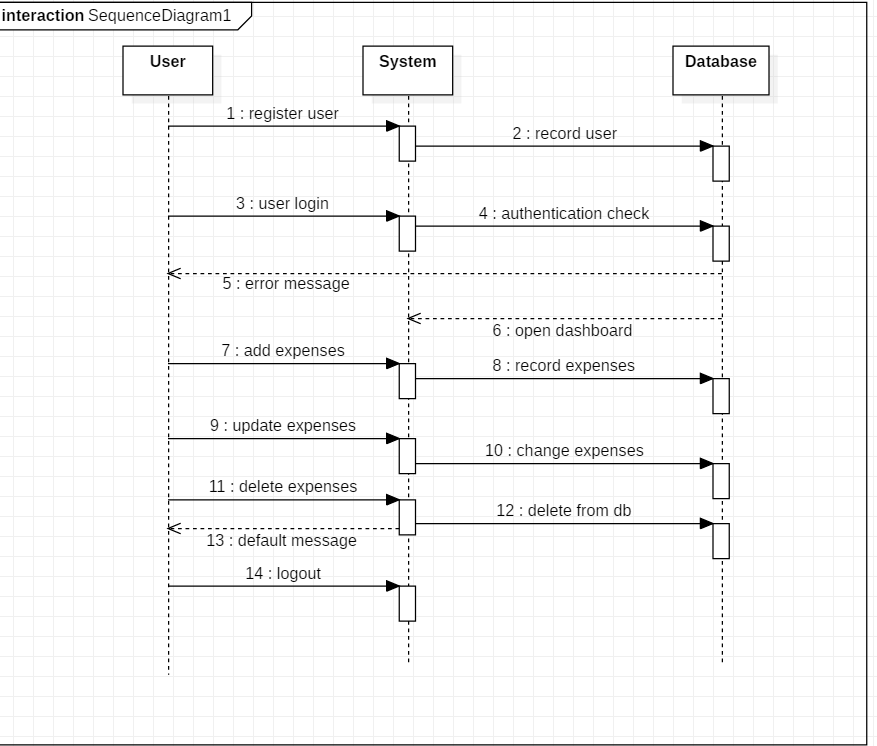
Justification:

First of all, the initialization is done by the user through registration process. Before entering into the system user must register data about them. The registered data will be stored into the user database. After registration user will be able to login but before logging into the dashboard there is authentication check which evaluates the username and password of user in the user database. Correct username and password will lead to the dashboard form where user can add, update, delete and search expenses which is recorded in expenses database.

Notation:

1. The start symbol: the black dot symbol/notation means the start of the system activity.
2. Decision symbol: the diamond shaped symbol is for the decision where user has to decide upon what to do next and it will perform accordingly.
3. Activity symbol: this symbol denotes the ongoing activity.
4. Object flow: It’s a line which denotes how the activity is going on.
5. Fork symbol: it is a shaded black block from where 3 or more flows are connected to activity symbol.
6. Sequence

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.



Justification:

Here in sequence diagram 3 different lifelines are made i.e. user system and database. Basically, it starts with the user action where (1) user registers the data into the system which further directs (2) to the database. (3) User then can login into the system but there is (4) an authentication check. If the username or password is not matched with the database it sends (5) error message to the user. If it matches then (6) dashboard form will open. In this form user can (7) add expense which further goes into the database (8). User can also update (9)(10) and delete (11) (12). If they don’t want to delete then it can also be done (13). User can logout (14)

1. Database design
2. Data dictionary

A data dictionary is a file or a set of files that contains a database's metadata. The data dictionary contains records about other objects in the database, such as data ownership, data relationships to other objects, and other data. The data dictionary is a crucial component of any relational database. Ironically, because of its importance, it is invisible to most database users. Typically, only database administrators interact with the data dictionary.

1. ERD

Entity Relationship Diagram, also known as **ERD**, ER Diagram or ER model, is a type of structural diagram for use in database design. An **ERD** contains different symbols and connectors that visualize two important information: The major entities within the system scope, and the inter-relationships among these entities.

1. Architectural design
2. UI design

User interface design or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. **User interface** (**UI**) **design** is the process of making interfaces in software or computerized devices with a focus on looks or style. Designers aim to create **designs** users will find easy to use and pleasurable. **UI design** typically refers to graphical user interfaces but also includes others, such as voice-controlled ones.

1. Prototyping

A **prototype** is an early sample, model, or release of a product built to test a concept or process or to act as a thing to be replicated or learned from. It is a term used in a variety of contexts, including semantics, design, electronics, and software programming.