**SOFTWARE ENGINEERING**

**(IT-314)**

**EVENT PLANNER APPLICATION**

Low Level Design Document

Version 1.0

**Team no: 14**

**Instructor: Prof. Asim Banerjee**

**Team Mentor: K Hemantha**

**Date: 29 February, 2016**

**Team Members**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name** | **ID** |
| 1. | Patel Yash | 201301008 |
| 2. | Patel Tej Naginbhai | 201301018 |
| 3. | Anavadiya Yash Kumar | 201301052 |
| 4. | Chauhan Dwimitra Dharmendra | 201301079 |
| 5. | Rahul Shashank Saranjame (Team Leader) | 201301101 |
| 6. | Fatnani Varun Kumarbhai | 201301152 |
| 7. | Nikita Jain | 201301155 |
| 8. | Karri Kiran Satish Reddy | 201301168 |
| 9. | Raman Kumar Douchaniya | 201301202 |

**Version History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version no.** | **Authors** |
| 29/02/2016 | 1.0 | ER diagram- Rahul Saranjame  Activity Diagram- Yash Anavadiya  Data Flow diagram- Kiran Reddy, Raman kumar  Sequence Diagram- Dwimitra Chauhan |

**Table of Contents**

1. Design Overview....................................................................................................4

1.1 Activity Diagrams………………………………………………………4

1.2 ER-Diagrams…………………………………………………………...12

1.3 Data Flow Diagrams…………………………………………………....13

1.4 Sequence Diagrams…………………………………………………….19

2. References……….………….................................................................................29

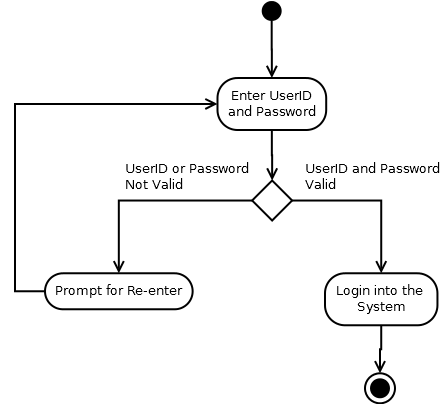
**1. Design Overview:**

**1.1 Activity Diagram:**

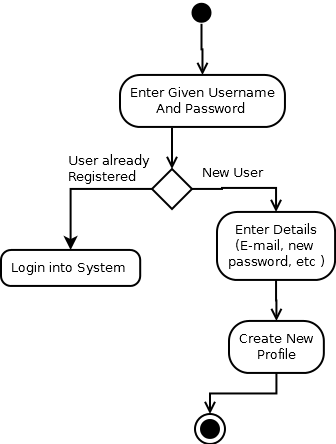
Activity Diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control is drawn from one operation to another. This flow can be sequential, branched or concurrent based on the dependencies of one activity on another.

The basic purposes of activity diagrams is the dynamic behavior of the system. Other diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another. Activity is a particular operation of the system. It shows control flow from one activity to another.

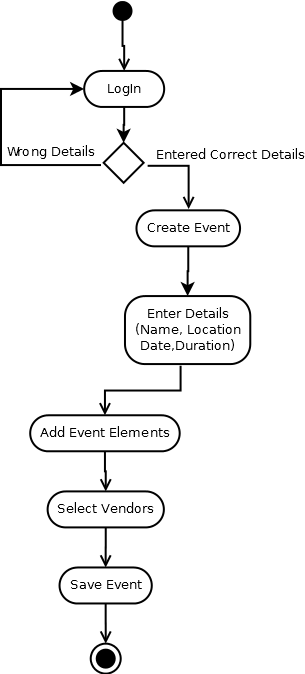
**Login:**



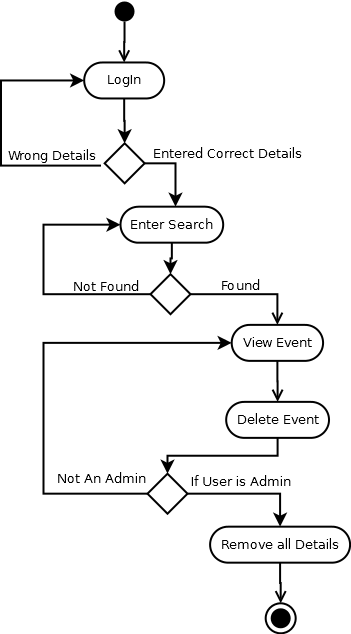
**Registration**

****

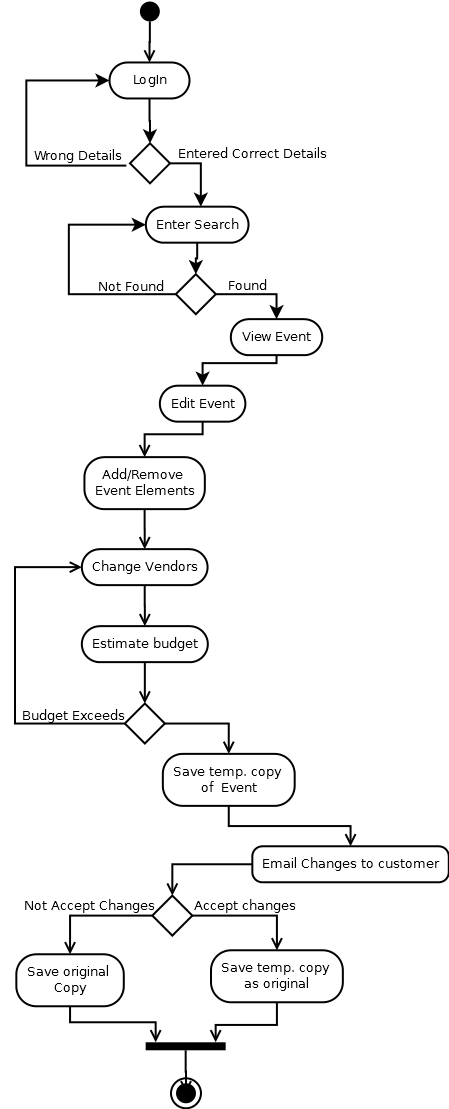
**CreateEvent**



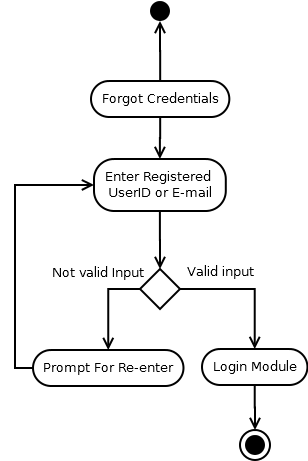
**Delete Event**

****

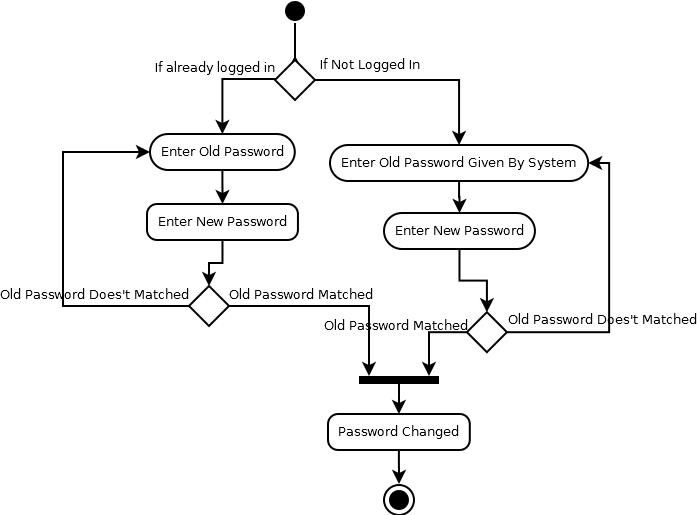
**Edit Event**

****

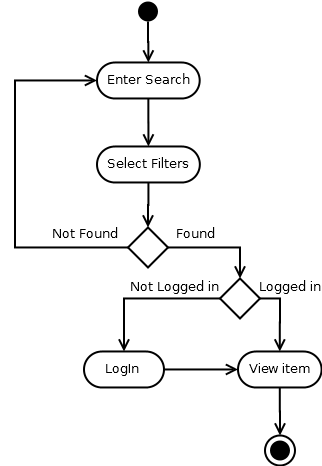
**Forgot Password**



**Change Password:**

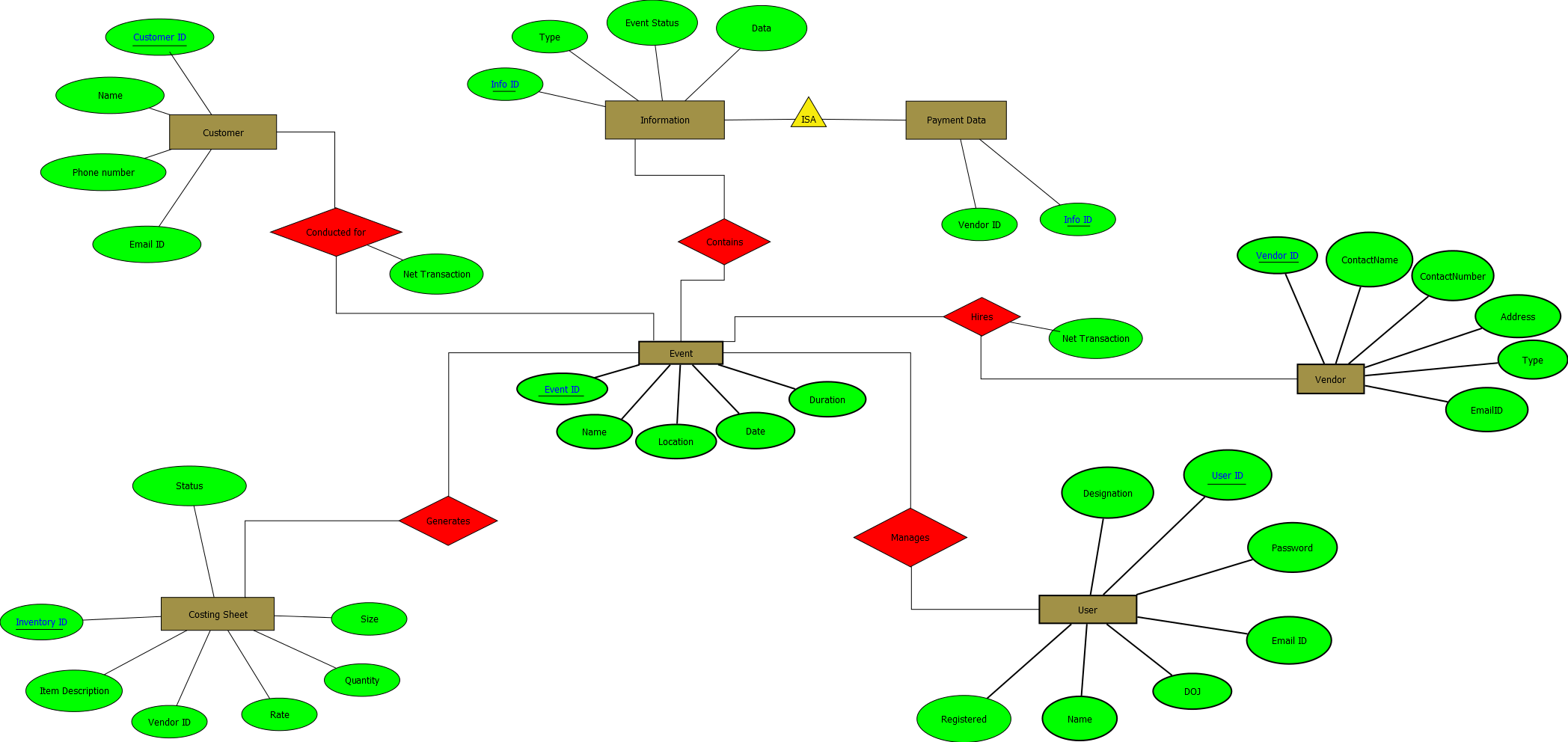
****

**Event Search:**

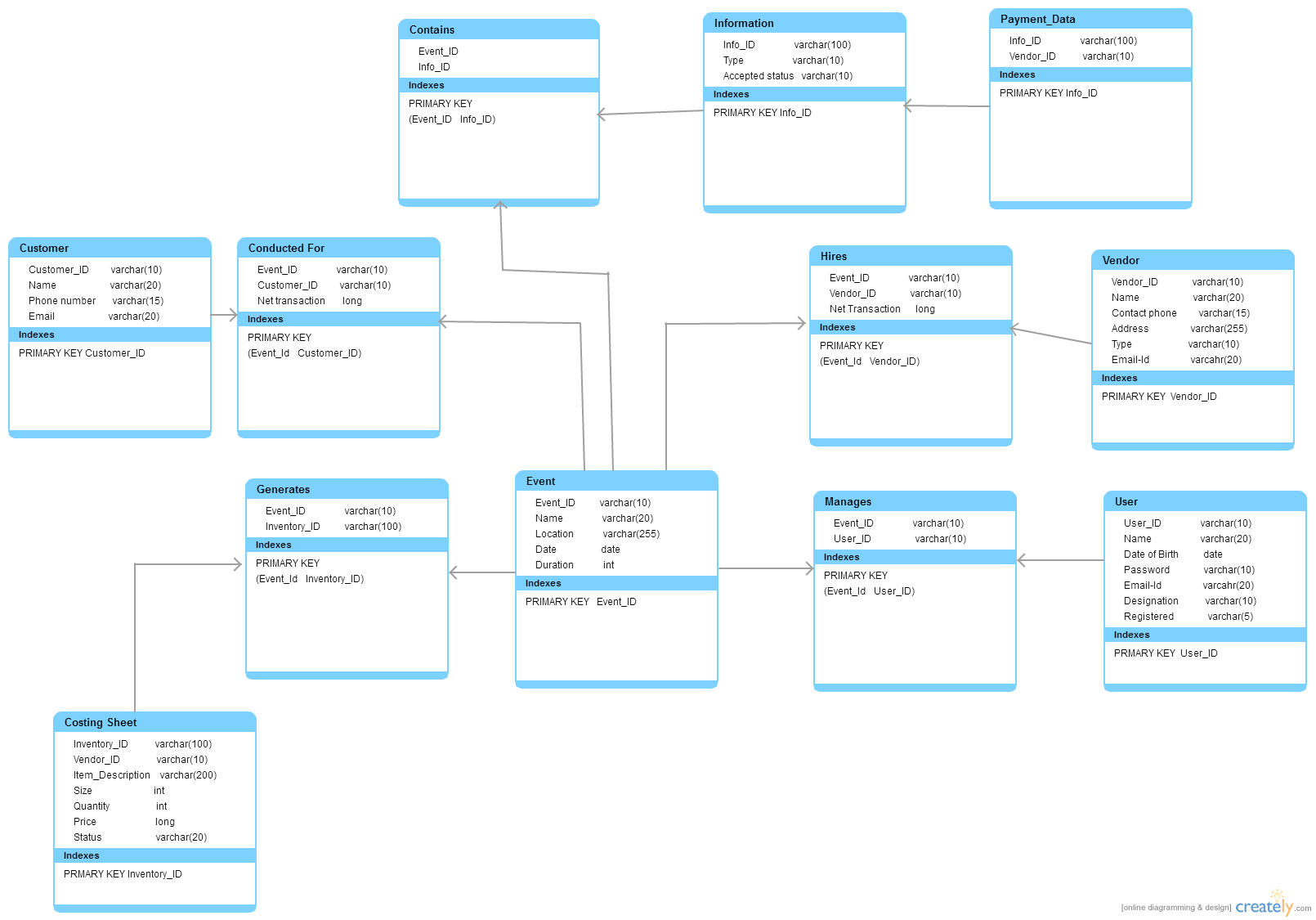


**1.2 ER-diagrams** :

Entity Relationship diagram is used to show the relation between different entities and each entity has at least one attribute. ER diagram is used to represent all the data associated with the software in a graphical format. It shows the relationships and dependencies between various data items of the database. Each entity has a unique id called primary key. There may be one more than one attribute acting as primary key. The relation may or may not have an attribute. There can be hierarchical entities deriving from an entity.



**1.2.1 Data Dictionary:**



**1.3 Data Flow Diagrams (DFD)**

Data Flow Diagram is graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. It represents the interaction of the system with database for retrieving data for achieving output for various activities by the user. It shows the flow of data from external entities into the system, flow of data from one module to another and flow of data to and fro between the logical storage and system modules.

Data Flow Diagrams are of two types:

* Logical Data Flow Diagram:

This type of DFD concentrates on the system process, and flow of data in the system

* Physical Data Flow Diagram:

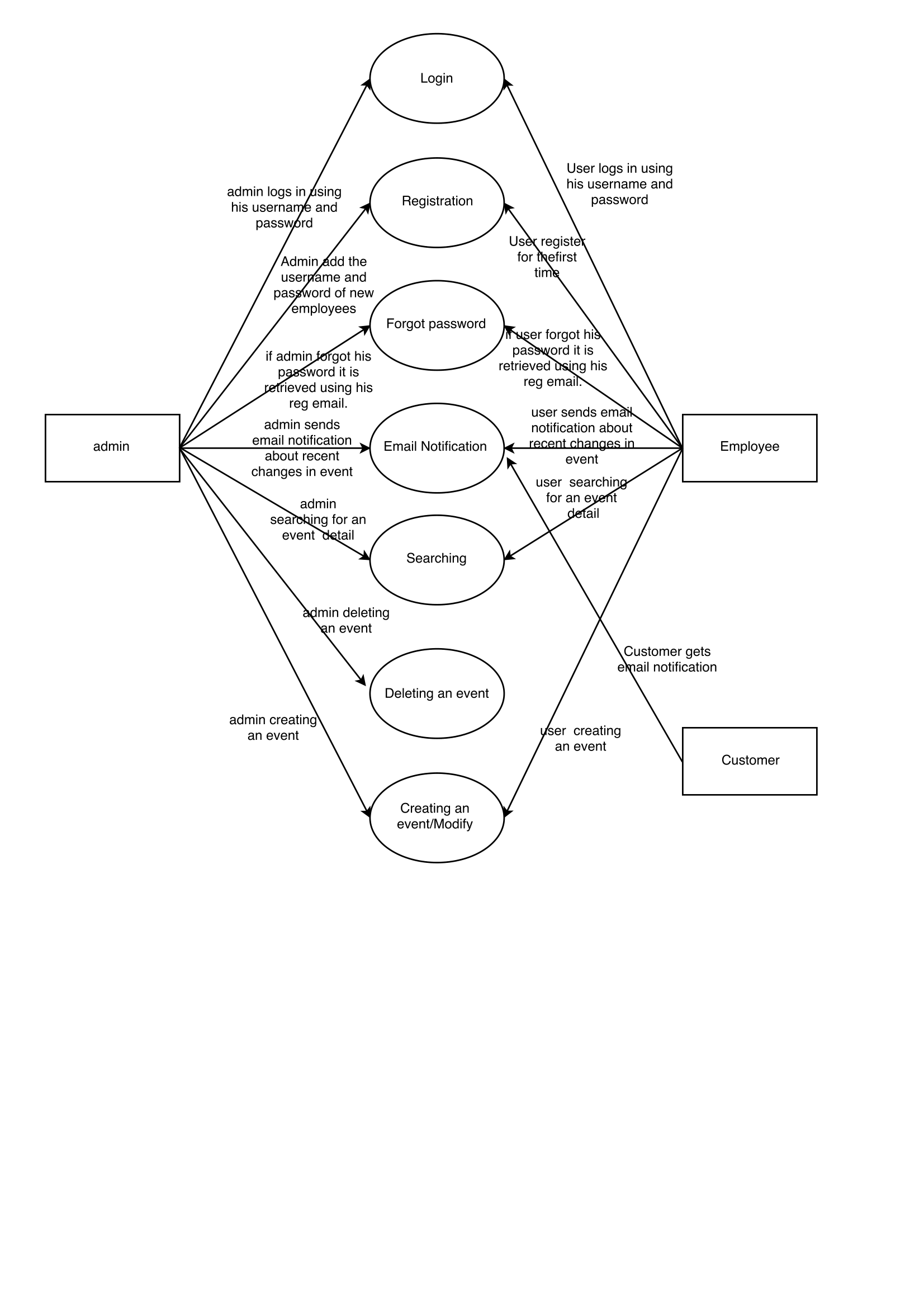
This type of DFD shows how the data flow is actually implemented in the system.It is more specific and close to the implementation. It shows how various functionalities are performed and who does them.

**Levels of Data Flow Diagram:**

* Level 0: Highest abstraction level DFD is known as Level 0 DFD, which depicts the entire information system as one diagram concealing all the underlying details.
* Level 1: The Level 0 DFD is broken down into more specific level i.e. Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules.

These two levels are sufficient for our project because every module can be clearly designed with in Level 1 phase. If we go one level ahead, it will be detailing an already detailed system. Therefore breaking down into further levels is not necessary because we will get whole idea of the module in Level 1 itself.

**Level 0 :**



**Level 1** :

**Login**

login.png

**Registration**

registration.png

**Forgot Password**

forgot_password.png

**Searching an Event**

searching.png

**Event Creation**

event creation.png

**Email Notification**

email_notification.png

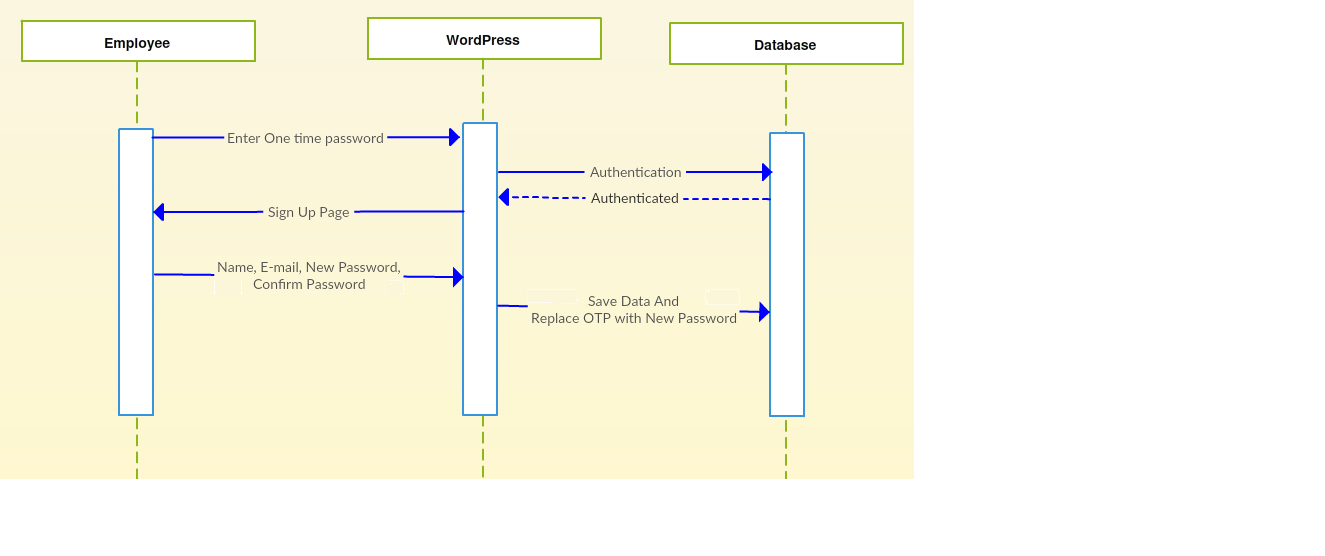
**Deleting an Event**

deleting an event.png

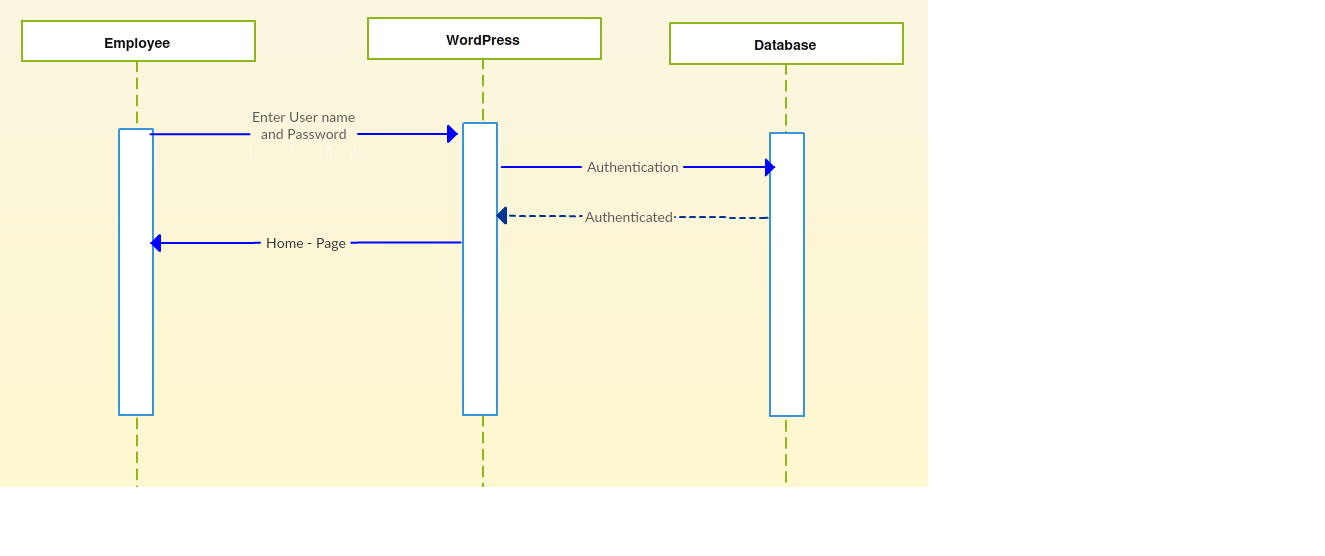
**1.4 Sequence Diagram:**

UML sequence diagram model the flow of logic within our system in a visual manner, enabling us to both to document and validate our logic that we commonly used for both analysis and design purpose. Sequence Diagram are the most useful artifact for dynamic modeling which focuses on identifying the behaviour of system modules with one another, with system backend and in what order. This constitutes the behaviour of overall system. Sequence Diagrams are typically used to model use case scenarios, the logic of methods, the logic of sequential flow in different modules.

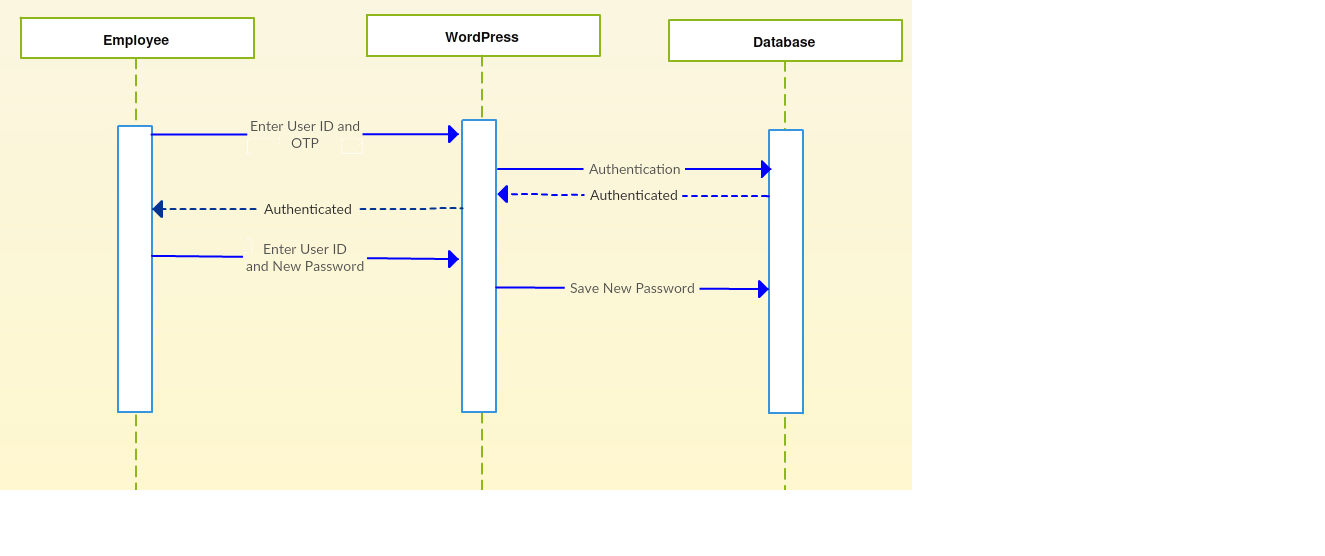
**1). Registration**

****

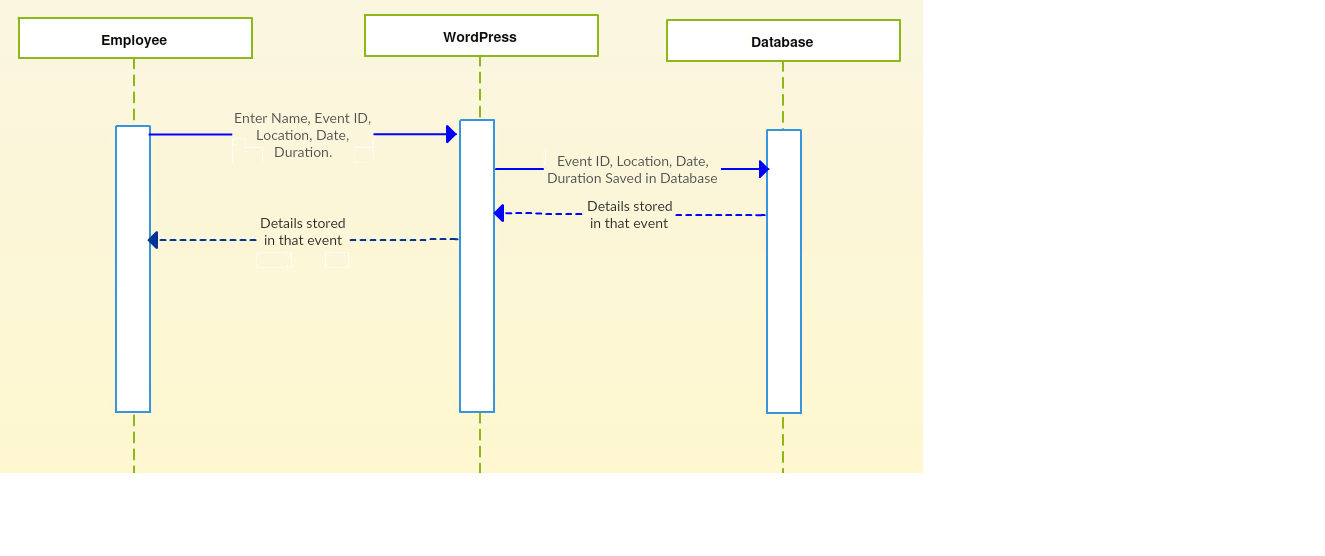
**2). Login**

****

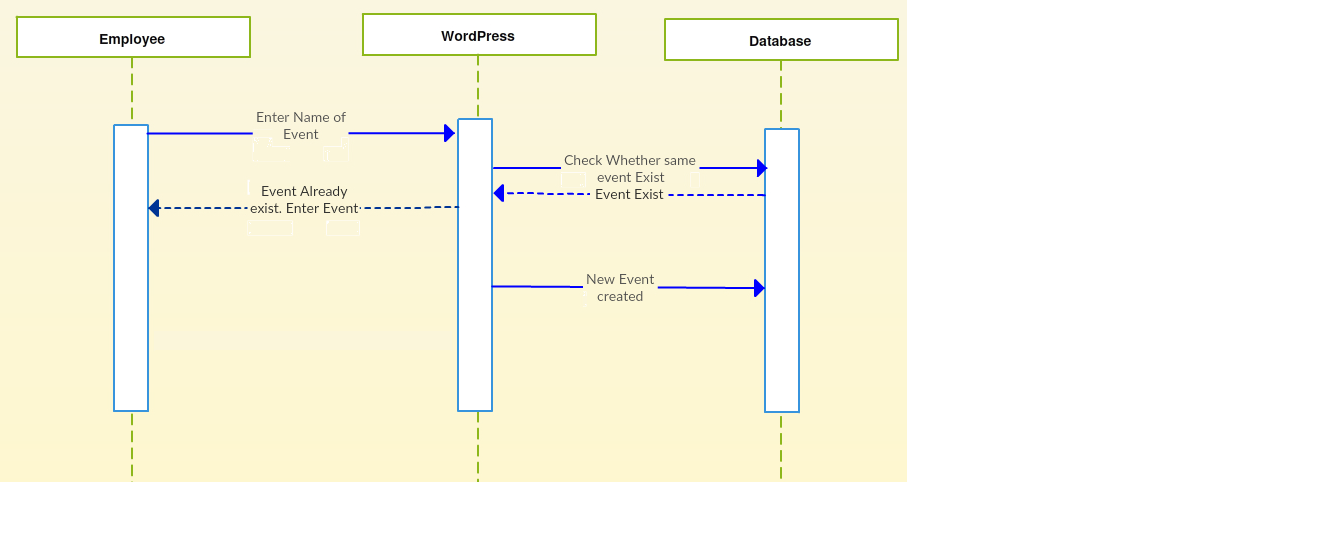
**3). Change Password**

****

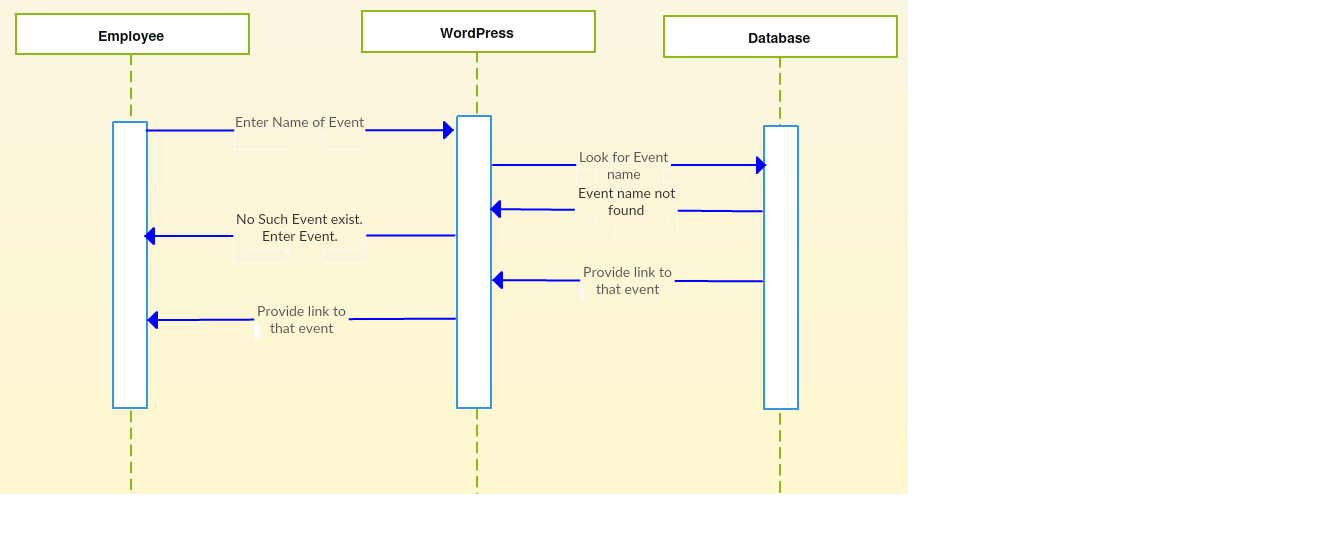
**4). Event Details**

****

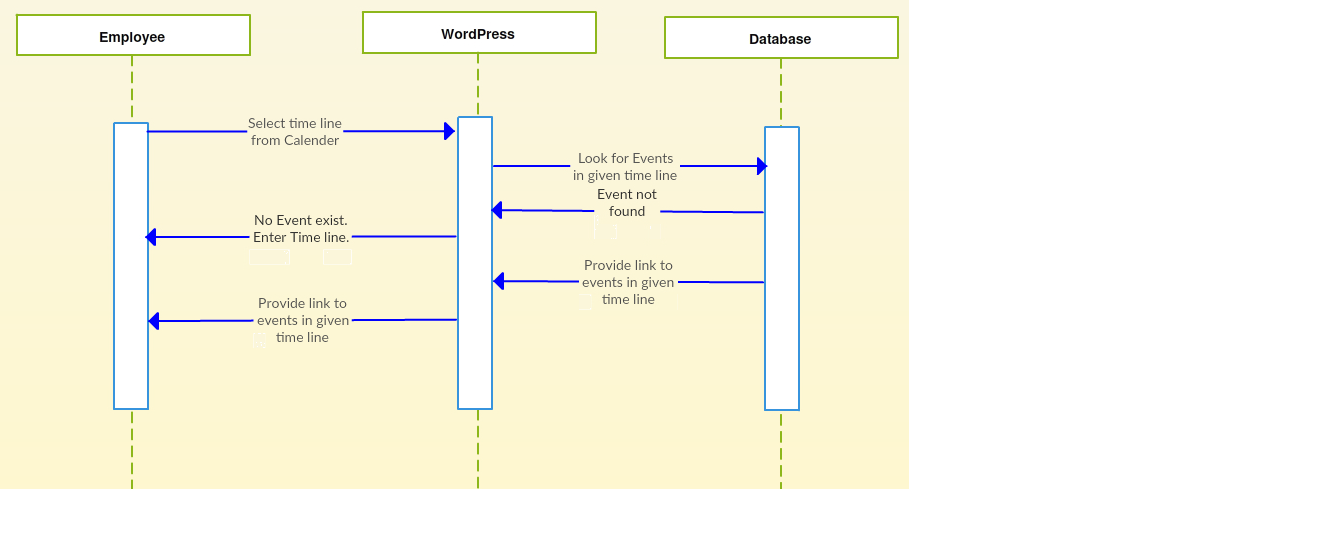
**5). Create Event**

****

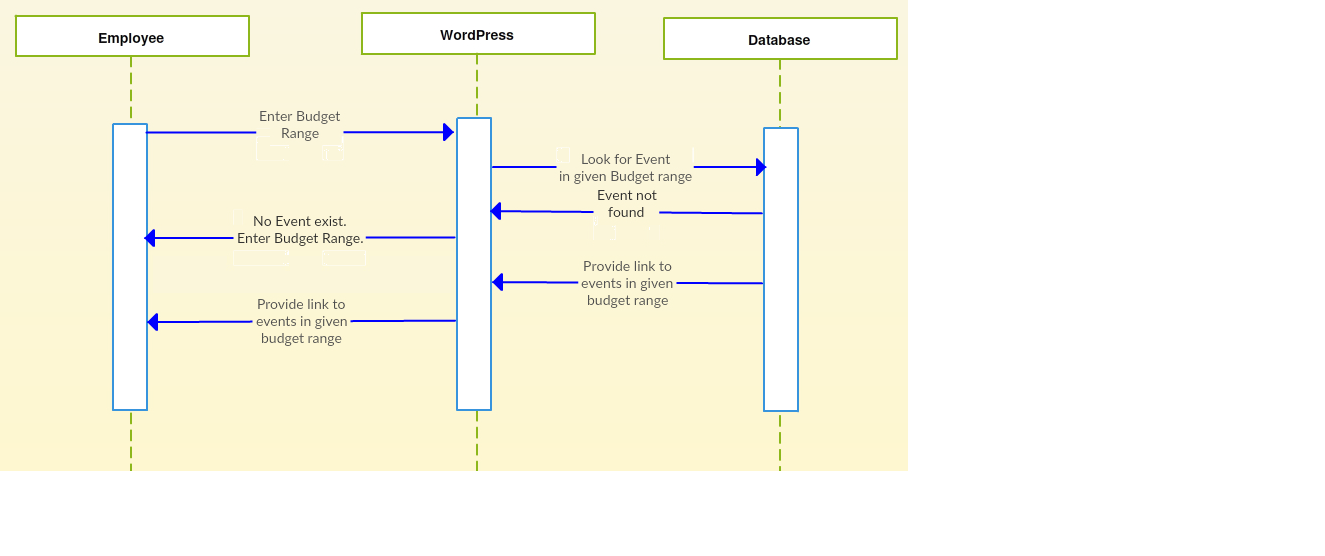
**6). Name Search**

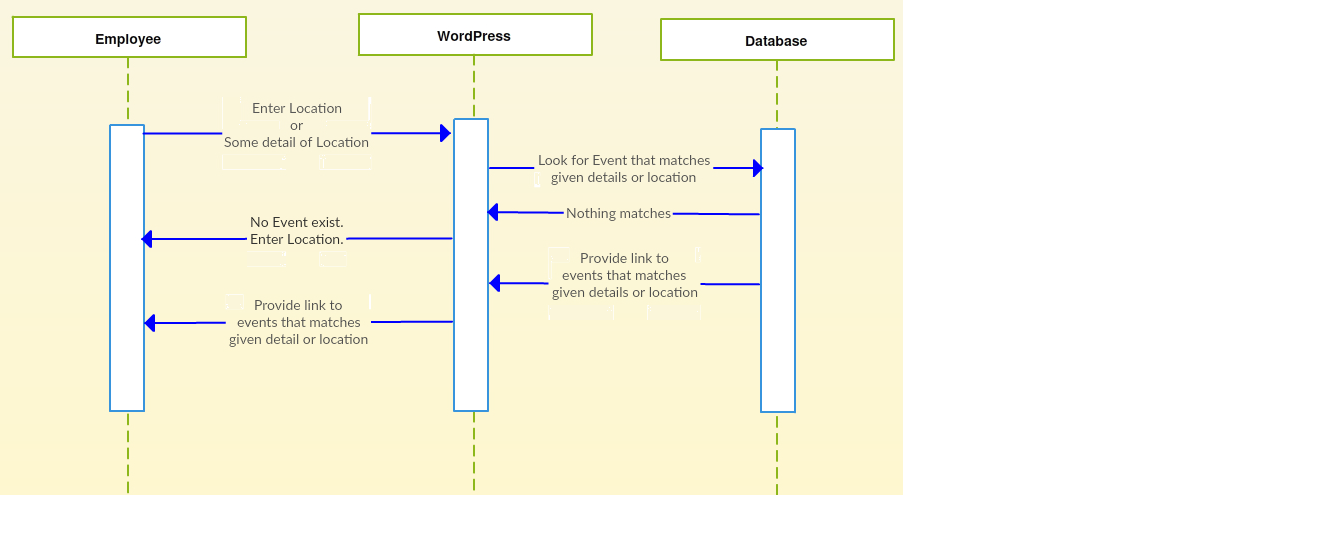
****

**7). Date Search**

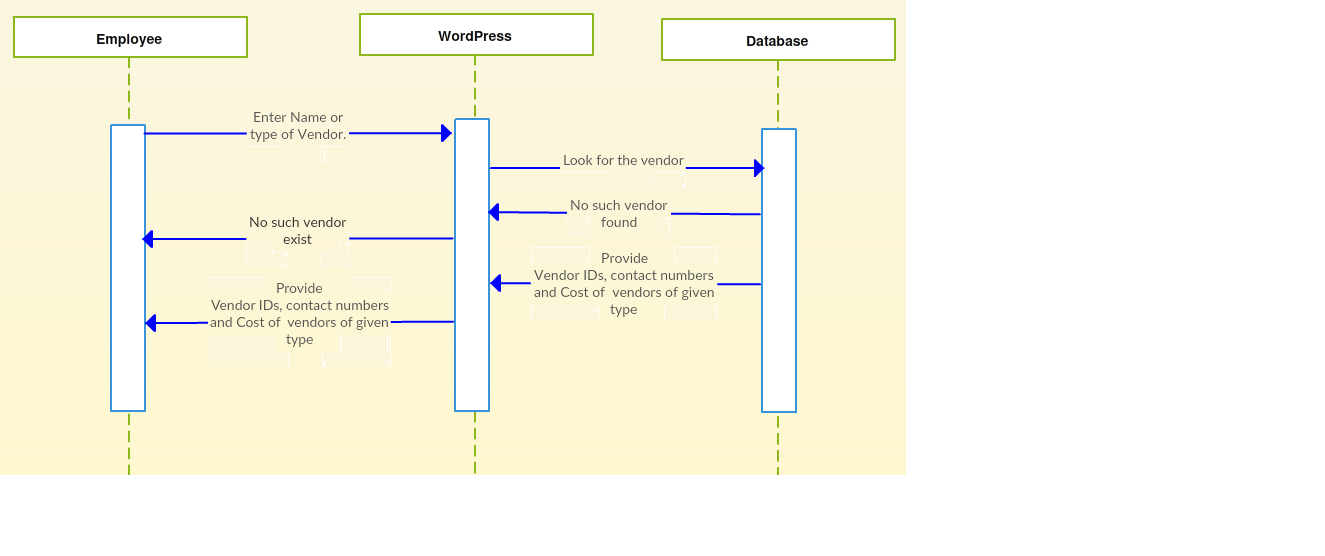
****

**8). Budget Search**

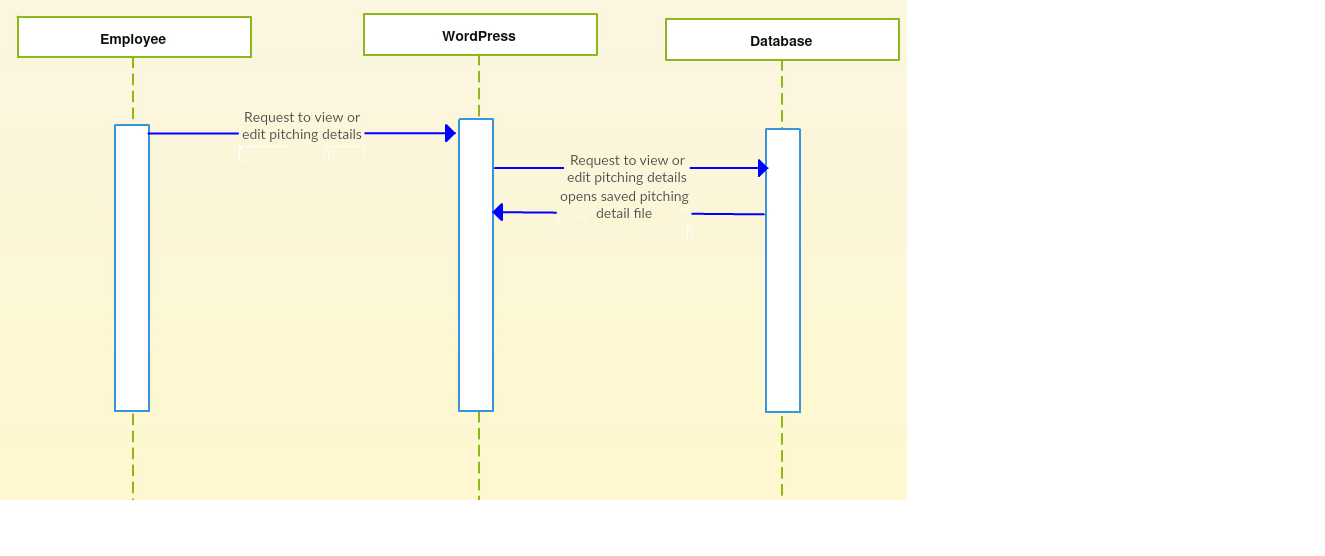
****

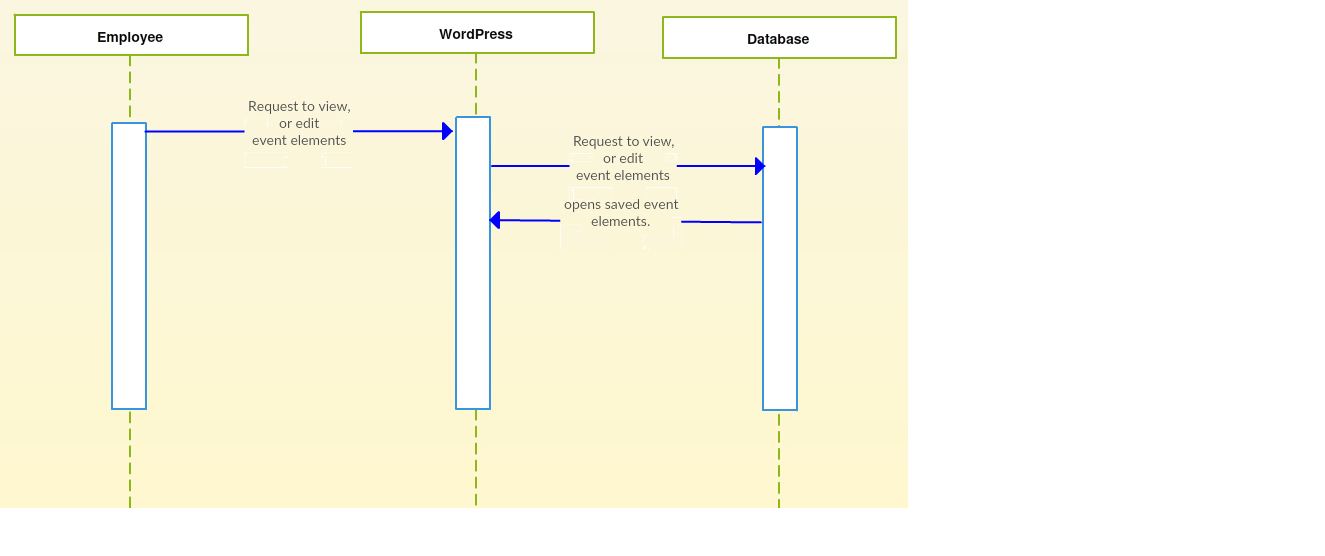
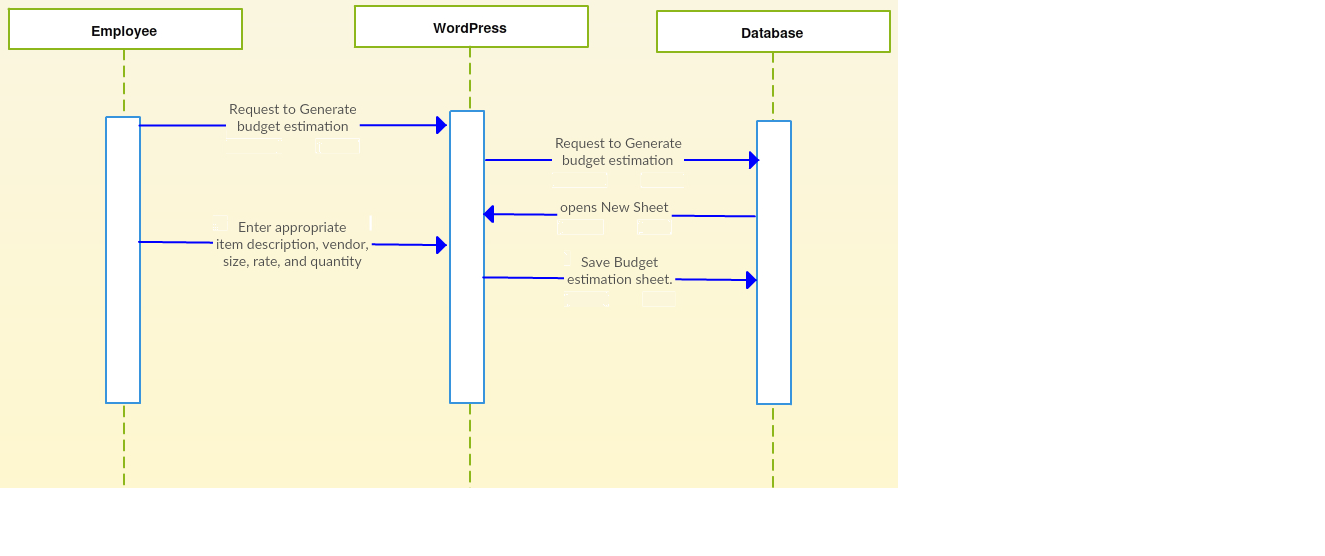
**9). Location Search**

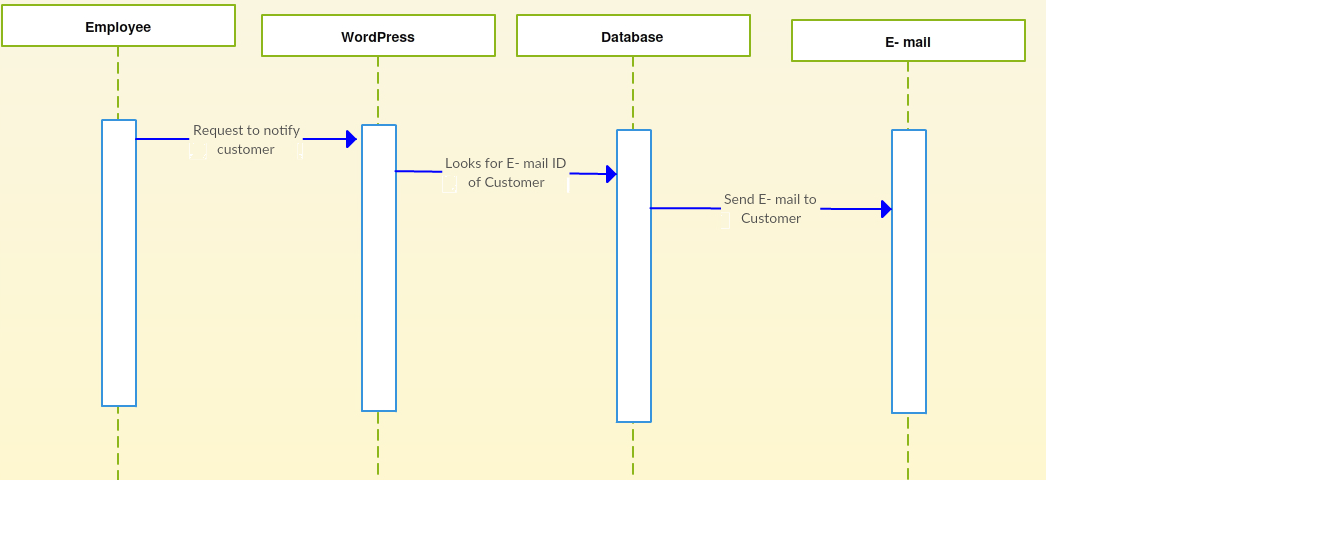
**10). Vendor Search**

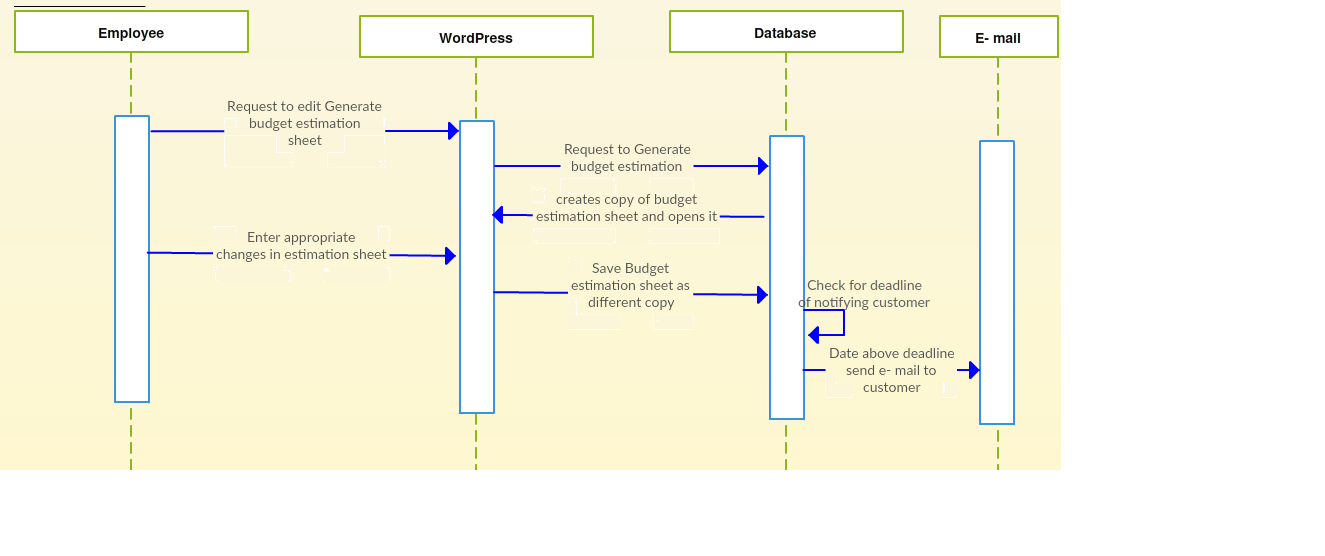
****

**11). Pitching Details**

****

**12). Event Elements/Vendor details13). Generate cost estimates by costing sheet prepared:**

**14). Notification**

**15). Cost Generation4. References :**

1. Low level design document v1.0, Team 7, IT314 Software Engineering, Winter 2012-13, DA-IICT.
2. Software Requirements Specification, Team 14, IT314 Software Engineering, Winter 2015-16, DA-IICT
3. <https://people.ok.ubc.ca/rlawrenc/research/Students/CJ_05_Design.pdf>
4. https://www.cms.gov/research-statistics-data.../**high**lvltechdesign.**doc**x