**Product Name**

MEEP

*Making Events Even Possible*

**Functional Requirements**

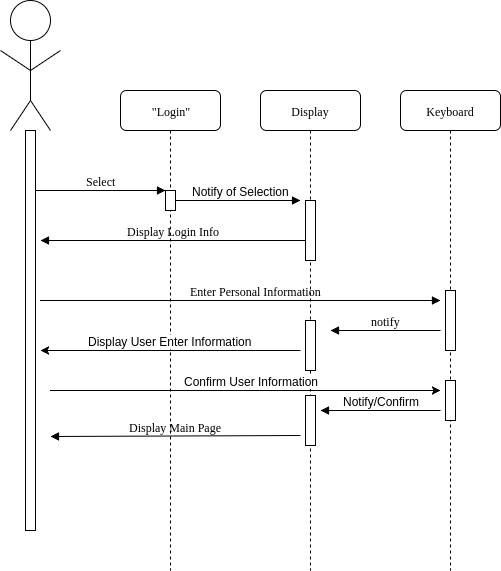
A single *User* has the ability to create, edit and view *Events*. A single event contains information pertaining to the event such as date, time, location and description. Before they can access ability to create events, they must log in as described in the sequence diagram. The two classes translate to databases of Users and Events. Events contain their own primary key as well as the foreign key of the creator of the event (User). There is an additional table which shows the many to many relationship between users. Many users can be friends with many users. The table contains a primary key for the entry, and two foreign keys for the relationship between users.

**Class Diagram**The three functional requirements all share the same data members and related functions, therefore there is single class diagram to represent them.

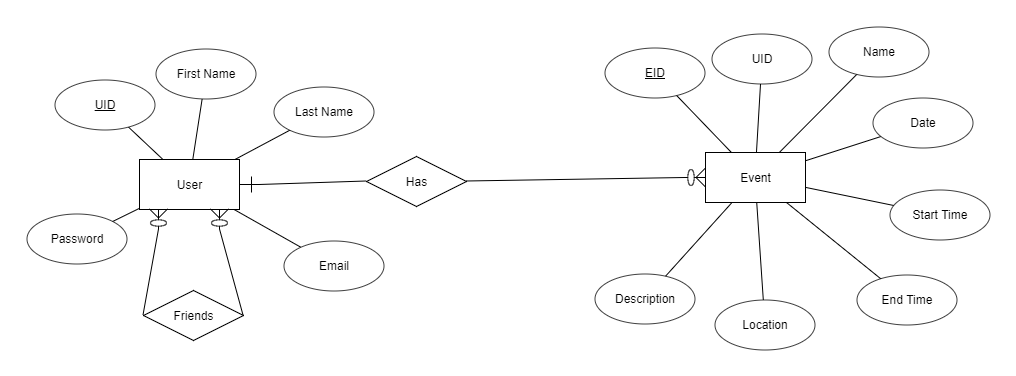
1. createEvent()
2. editEvent()
3. viewDay()

|  |  |  |
| --- | --- | --- |
| **User** |  | **Event** |
| -UID: string // user ID  -FirstName: string  -LastName: string  -Username: string  -Email: string  -Password: string | 1 create/edit/view 0\*  user event | -EID: string // event ID  -UID: string // FK  -Name: string  -Date: string  -StartTime: string  -EndTime: string  -Location: string  -Description: string  -Style: Display |
| setUID()  getUID()  setFirstName()  getFirstName()  setLastName()  getLastName()  setUsername()  getUsername()  setEmail()  getEmail()  setPassword()  getPassword() |  | setEID()  getEID()  getUID()  setName()  getName()  setDate()  getDate()  setStartTime()  getStartTime()  setEndTime()  getEndTime()  setLocation()  getLocation()  setDescription()  getDescription()  setStyle()  getStyle() |

**Sequence Diagram**

****

**Entity Relational Diagram**

****

**Relational Diagram**

