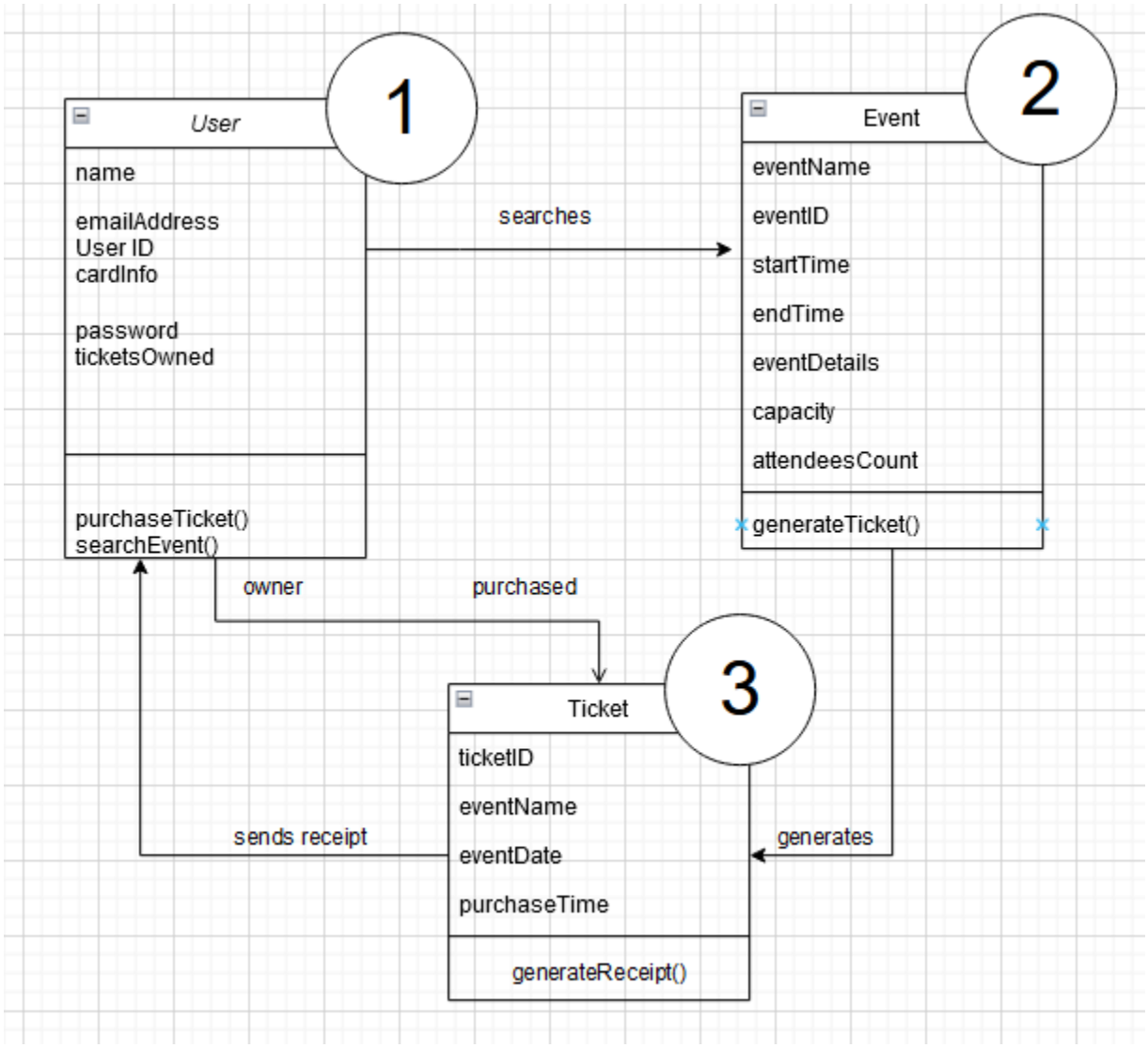
Suggested Object Responsibilities

Object	**Responsibilities (Knowing)**	**Responsibilities (Doing)**
User	User’s name User ID User’s email address Current Tickets	Purchasing a ticket Searching for an event
Event	Event name Event ID Date of event Attendant capacity and count	Generating a ticket after purchase for the user to own.
Receipt	Date of purchased ticket Event details Owner of Receipt	Send a receipt to the user's email.
Ticket	Ticket ID Event name Date of event	Generates a receipt for the user.

Implementation Order Diagram



Objects And Operations

```
class WebServer <singleton>
      private UserDatabase userDB;
      private EventDatabase eventDB;
      private TicketDatabase ticketDB;

      public constructor(
            String udbPath,
            String edbPath,
            String tdbPath,
```

```
	)

	public respondToGETRequest(//event handler
		string url,
		string params,
		string cookies/login/etc.
	)

	public respondToPOSTRequest(//event handler
		string url,
		string data
	)

	event[] queryEvents(string Query)

	bool purchaseTicket(
		int uid,
		int ticketid,
		Card paymentInfo
	)

	//document does not specify that each event must have a unique
	cost, so $10 will be charged for all tickets
	bool chargeCard(card paymentInfo);

	bool createUser(string name,string email,string password)

	bool sendEmailReceipt(Ticket ticket)

class User:
	int userID;
	String name;
	String passwordHash;
	String passwordSalt;
	String email;
	Ticket tickets[];

	constructor(string name,string email,string password)
	static validatePassword(string p)
```

```
class Event:
      int eventID;
      int capacity;
      int remainingSeats;
      DateTime startTime;
      DateTime endTime;
      string eventName;
      string eventDetails

      constructor(
            String eventName,
            int capacity,
            DateTime startTime,
            DateTime endTime
      );

      updateDetails(String details);


class Ticket:
      User attendee;
      Event event;
      DateTime purchaseTime;
      generateReceipt();

      constructor(User user, Event event)
      constructor(User user, Event event, DateTime PurchasedOn)

class UserDatabase:
      constructor(string path) //read db from disk, create objects
      addUser(User u)
      writeToDisk()
class EventDatabase:
      constructor(string path) //read db from disk, create objects
      addEvent(Event e)
      writeToDisk() //write db to disk
class TicketDatabase:
      constructor(string path) //read db from disk, create objects
```

```
addTicket(Ticket t)
writeToDisk()
```