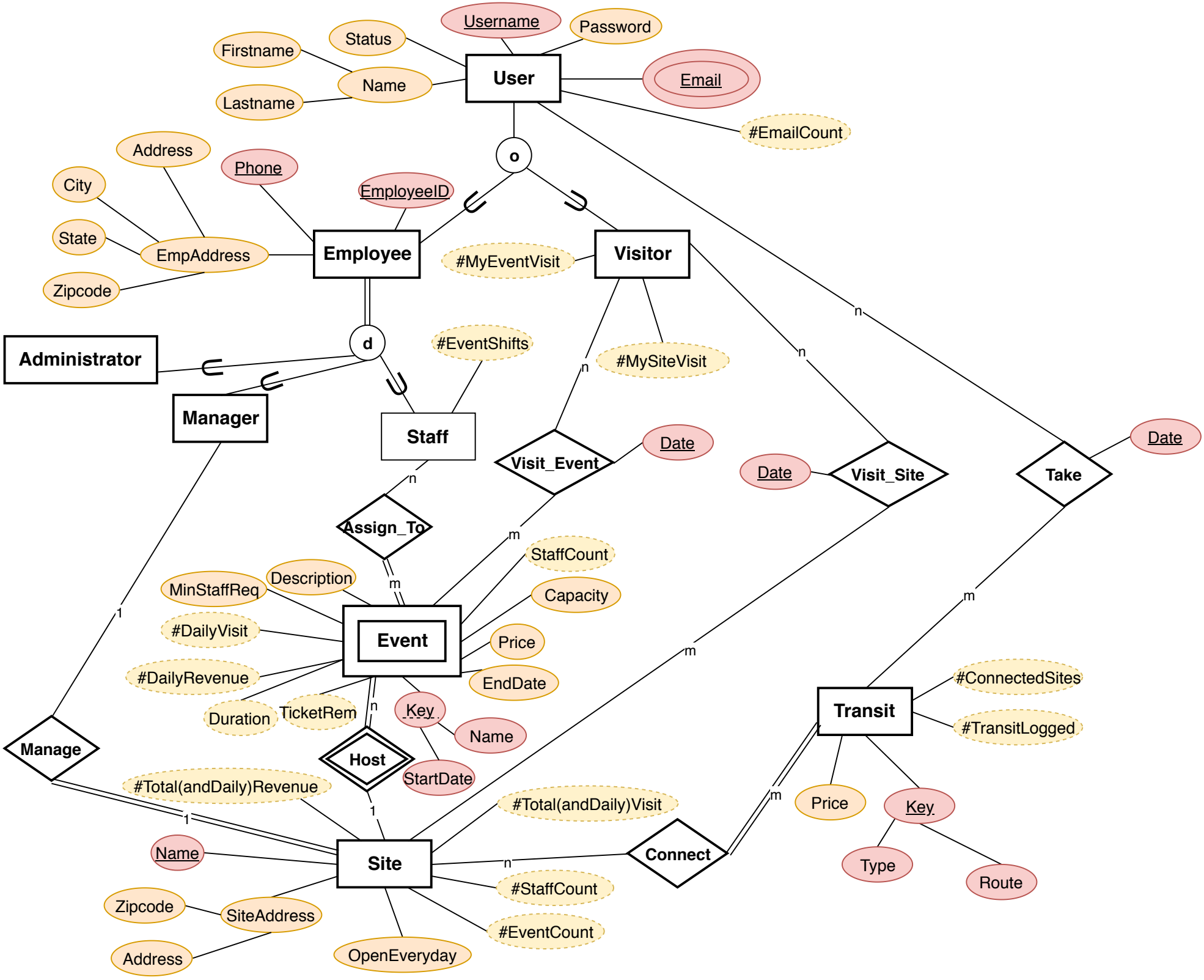
TEAM 6

Phase 2

Nick Loprinzo (nloprinzo3)
Kristof Kovacs (kkovacs3)
Junwoo Park (jpark759)
Anand Sarathy (asarathy7)
Chase Warner (cwarner30)


Username
Password
Status
Firstname
Name
Lastname
User
Email
#EmailCount
o
Address
Phone
City
EmployeeID
State
EmpAddress
Employee
#MyEventVisit
Visitor
Zipcode
d
#EventShifts
#MySiteVisit
Administrator
Manager
Staff
Visit_Event
Date
Date
Visit_Site
Take
Date
n
n
n
n
Assign_To
m
m
m
m
1
1
1
StaffCount
Description
MinStaffReq
Capacity
#DailyVisit
Event
Price
#DailyRevenue
EndDate
Duration
TicketRem
n
Key
Name
Manage
Host
StartDate
#Total(andDaily)Revenue
#Total(andDaily)Visit
Transit
#ConnectedSites
#TransitLogged
Price
Key
Name
Site
Connect
Type
Route
#StaffCount
Zipcode
SiteAddress
#EventCount
Address
OpenEveryday

Relational Schema Diagram
Primary Key

User = (Username, Password, FirstName, LastName, UserStatus)

UserEmails = (Username[fk1], Email)
fk1: Username → User.Username

Employee = (Username[fk2], EmployeeID, Phone, State, City, Zipcode, Address, isAdministrator, isManager, isStaff)
fk2: Username → User.Username

Visitor = (Username[fk3])
fk3: Username → User.Username

Transit = (Transit_type, Route, Price)

Site = (Name, ManagerUsername[fk4], Zipcode, Address, OpenEveryday)
fk4: ManagerUsername → Employee.Username

Event = (Name, StartDate, SiteName[fk5], Capacity, Price, EndDate, MinStaffReq, Description)
fk5: SiteName → Site.Name

VisitEvent = (VisitDate, VisitorUsername, EventName, EventStartDate[fk6], SiteName[fk7])
fk6: VisitorUsername → Visitor.Username
fk7: EventName, EventStartDate, SiteName → Event.Name, Even.StartDate, Event.SiteName

EventStaff = (EventName, EventStartDate, SiteName[fk8], StaffUsername[fk9])
fk8: EventName, EventStartDate, SiteName → Event.Name, Event.StartDate, Event.SiteName
fk9: StaffUsername → Employee.Username

TransitConnectsSites = (SiteName[fk10], TransitType, TransitRoute[fk11])
fk10: SiteName → Site.Name
fk11: TransitType, TransitRoute → Transit.Transit_Type, Transit.Route

VisitSite = (VisitDate, VisitorUsername[fk12], SiteName[fk13])
fk12: VisitorUsername → Visitor.Username
fk13: SiteName → Site.Name

UserTakeTransit = (Transit_Date, Username[fk14], Transit_Type, Route[fk15])
fk14: Username → User.Username
fk15: Transit_Type, Route → Transit.Transit_Type, Transit.Route

MySQL Create Table Statements

```
#Drop table if exists `UserTakeTransit`;
#Drop table if exists `VisitSite`;
#Drop table if exists `TransitConnectsSites`;
#Drop table if exists `EventStaff`;
#Drop table if exists `VisitEvent`;
#Drop table if exists `Event`;
#Drop table if exists `Site`;
#Drop table if exists `Transit`;
#Drop table if exists `Visitor`;
#Drop table if exists `Employee`;
#Drop table if exists `UserEmails`;
#Drop table if exists `User`;

Create Table User (
	Username varChar(63),
	UserPassword varChar(63) Not Null,
	FirstName varChar(31),
	LastName varChar(31),
	UserStatus varChar(7) Default 'P',
	Primary Key (Username),
	Constraint CK_Password_Length check (LEN(UserPassword) >= 8)
);

Create Table UserEmails (
	Username varChar(63),
	Email varChar(255) Not Null Unique,
	Constraint PK_UserEmail PRIMARY KEY (Username, Email),
	Constraint Valid_Email check (Email like "%[a-z0-9]%@%.%"),
	FOREIGN KEY (Username) REFERENCES User(Username)
		on update cascade
		on delete cascade
);

Create Table Employee (
	Username varChar(63) Primary Key,
	EmployeeID int(9) not null unique,
	Phone int(10) not null unique,
	State varChar(2) not null,
	City varChar(63) not null,
	Zipcode int(5) not null,
	Address varChar(200) not null,
	isAdministrator bit not null,
	isManager bit not null,
	isStaff bit not null,
	FOREIGN KEY (Username) REFERENCES User(Username)
		On Update cascade
		On Delete cascade
);
```

```
Create table Visitor (
	Username varchar(63) Primary Key,
	Foreign key (Username) references User(Username)
		On update cascade
		On delete cascade
);

Create Table Transit (
	Transit_type varchar(10),
	Route varchar(10),
	Price int not null,
	Check (Price >= 0),
	Constraint PK_transit Primary key (Transit_type, Route)
);

Create Table Site (
	Name varchar(63) Primary Key,
	ManagerUsername varchar(63) not null,
	Zipcode int(5) not null,
	Address varchar(200) not null,
	OpenEveryday bit not null,
	Foreign Key (ManagerUsername) references Employee(Username)
		On update cascade
		On delete restrict
);

Create Table Event(
	Name varChar(63),
	StartDate Date,
	SiteName varChar(63),
	Capacity int Not Null,
	Check (Capacity >= 0),
	Price int Not Null,
	Check (Price >=0),
	EndDate Date Not Null,
	MinStaffReq int Not Null,
	Check (MinStaffReq >=0),
	Description varChar(280),
	FOREIGN KEY (SiteName) REFERENCES Site(Name)
		On update cascade
		On delete cascade,
	Constraint PK_Event Primary Key (Name, StartDate, SiteName),
	Constraint Date_Event check(StartDate < EndDate)
);
```

```
Create table VisitEvent(
        VisitDate Date,
        VisitorUsername varchar(63),
        EventName varChar(63),
        EventStartDate Date,
        SiteName varChar(63),
        Constraint PK_VisitEvent Primary Key
                (VisitDate, VisitorUsername, EventName, EventStartDate, SiteName),
        Foreign Key (VisitorUsername) references Visitor(Username)
                On update cascade
                On delete cascade,
        Constraint FK_VisitEvent Foreign Key (EventName, EventStartDate, SiteName)
                References Event(Name, StartDate, SiteName)
                On update cascade
                On delete cascade
);

Create table EventStaff (
        EventName varChar(63),
        EventStartDate Date,
        SiteName varChar(63),
        StaffUsername varChar(63),
        Constraint FK_EventStaff Foreign Key (EventName, EventStartDate, SiteName)
                References Event(Name, StartDate, SiteName)
                On update cascade
                On delete cascade,
        Foreign Key (StaffUsername) References Employee(Username)
                On Update cascade
                On Delete Cascade,
        Constraint PK_EventStaff Primary Key (EventName, EventStartDate, SiteName)
);

Create Table TransitConnectsSites (
        SiteName varChar(63),
        TransitType varChar(10),
        TransitRoute varChar(10),
        Foreign Key (SiteName) references Site(Name)
                On update cascade
                On delete restrict,
        Constraint FK_Transits_Conn Foreign Key (TransitType, TransitRoute) References
                Transit(Transit_type, Route)
                On update cascade
                On delete cascade,
        Constraint PK_Transits_Conn Primary Key (SiteName, TransitType, TransitRoute)
);
```

```
Create table VisitSite (
        VisitDate Date,
        VisitorUsername varChar(63),
        SiteName varChar(63),
        Foreign Key (VisitorUsername) references User(Username)
                On update cascade
                On delete cascade,
        Foreign Key (SiteName) references Site(Name)
                On update cascade
                On delete cascade,
        Constraint PK_VisitSite Primary Key (VisitDate, VisitorUsername, SiteName)
);

Create Table UserTakeTransit (
        Transit_Date Date,
        Username varChar(63),
        Transit_type varchar(10),
        Route varchar(10),
        Foreign Key (Username) references User(Username)
                on update cascade
                on delete cascade,
        Constraint FK_UserTransit Foreign Key (Transit_type, Route) references Transit(Transit_type, Route)
                on update cascade
                on delete cascade,
        Constraint PK_UserTransit Primary Key (Transit_Date, Username, Transit_type, Route)
);
```