

## CS 301 Phase II - Group 2

Henry Bugajski - [hjbugajski@crimson.ua.edu](mailto:hjbugajski@crimson.ua.edu) (Section 002)

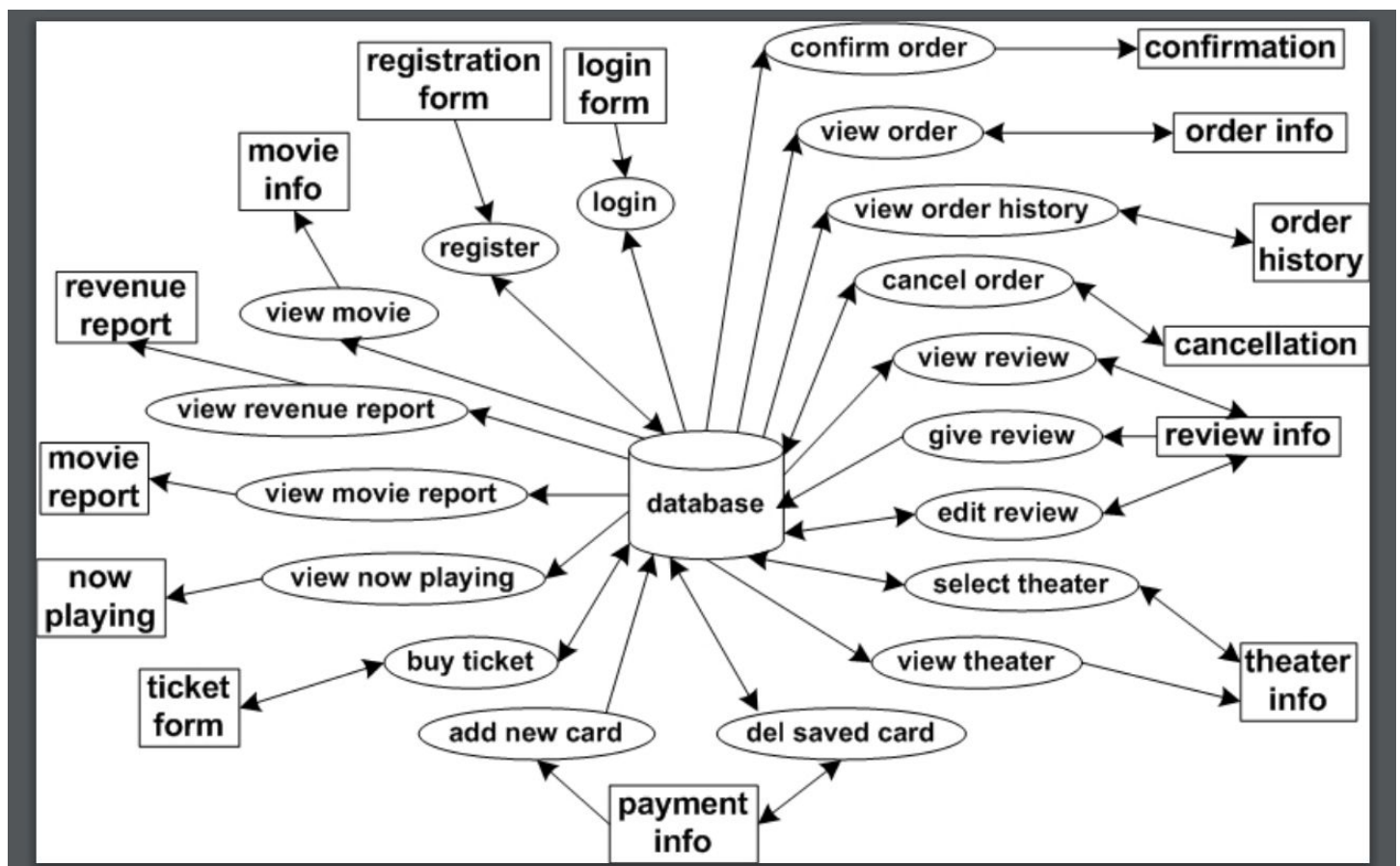
Logan Mimms - [lgmimms@crimson.ua.edu](mailto:lgmimms@crimson.ua.edu) (Section 002)

Chas Shipman - [ceshipman@crimson.ua.edu](mailto:ceshipman@crimson.ua.edu) (Section 002)

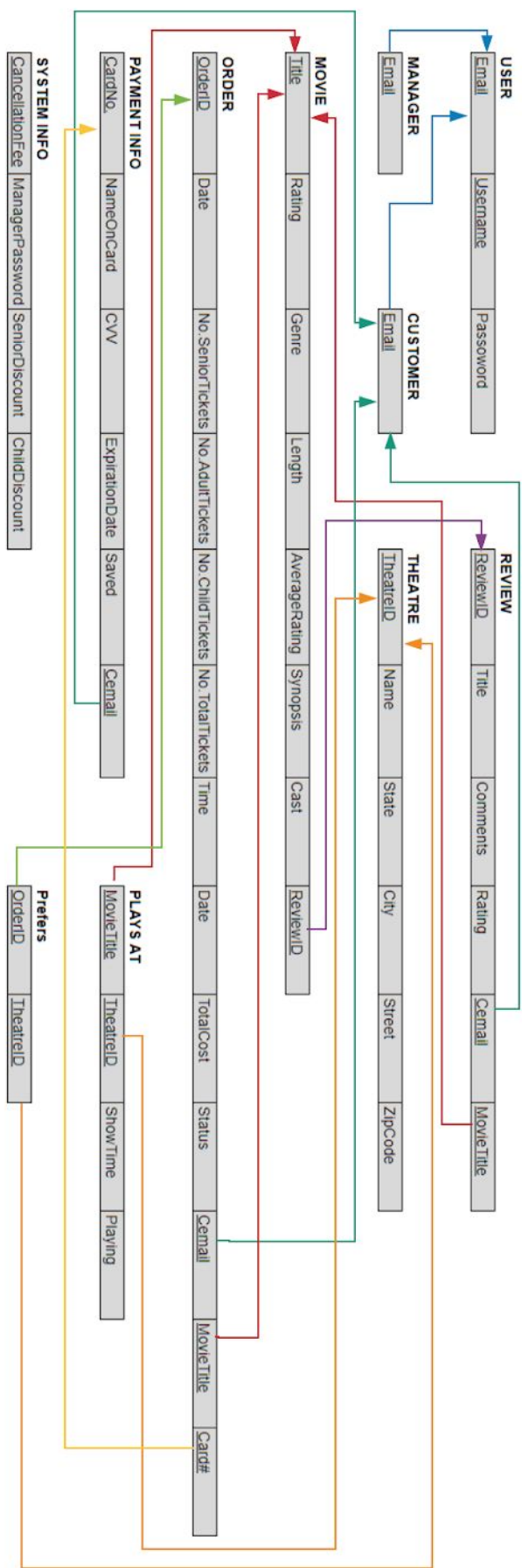
Minrui Huang - [mhuang11@crimson.ua.edu](mailto:mhuang11@crimson.ua.edu) (Section 002)



## Information Flow Diagram



Relational Schema Diagram



## Table statements (including domain constraints, integrity constraints, primary keys, and foreign keys)

```
CREATE TABLE `User` (  
    `Email` VARCHAR (50) NOT NULL UNIQUE,  
    `Username` VARCHAR(15) NOT NULL UNIQUE,  
    `Password` VARCHAR(30),  
    PRIMARY KEY(`Username`,`Email`)  
);  
  
CREATE TABLE `Manager` (  
    `Email` varchar (50) REFERENCES `User`(`Email`) NOT NULL,  
    PRIMARY KEY(`Email`)  
);  
  
CREATE TABLE `Customer` (  
    `Email` VARCHAR(50) REFERENCES `User`(`Email`) NOT NULL,  
    PRIMARY KEY(`Email`)  
);  
  
CREATE TABLE `Review` (  
    `ReviewID` INTEGER NOT NULL UNIQUE,  
    `Title` VARCHAR(30) NOT NULL,  
    `Comments` VARCHAR(500) NOT NULL,  
    `Rating` FLOAT CHECK(0 <= Rating <= 5) NOT NULL,  
    `Cemail` Varchar(50) NOT NULL UNIQUE REFERENCES `Customer`(`Email`),  
    `MovieTitle` VARCHAR(50) NOT NULL UNIQUE REFERENCES `Movie`(`Title`),  
    PRIMARY KEY(`ReviewID`)  
);  
  
CREATE TABLE `SystemInfo` (  
    `CancellationFee` INTEGER NOT NULL CHECK(CancellationFee >=0),  
    `ManagerPassword` INTEGER NOT NULL,  
    `SeniorDiscount` FLOAT NOT NULL,  
    `ChildDiscount` FLOAT NOT NULL,  
    PRIMARY KEY(`CancellationFee`)  
);  
  
CREATE TABLE `Theatre` (  
    `TheatreID` INTEGER NOT NULL,  
    `Name` VARCHAR(50) NOT NULL,  
    `State` VARCHAR(2) NOT NULL,  
    `City` VARCHAR(50) NOT NULL,  
    `Street` VARCHAR(50) NOT NULL UNIQUE,  
    `ZipCode` INTEGER NOT NULL CHECK(10000 <= ZipCode <= 99999),  
    PRIMARY KEY(`TheatreID`)  
);
```

```

CREATE TABLE `Movie` (
  `Title` VARCHAR(50) NOT NULL,
  `Rating` INT CHECK(0 <= Rating <= 5),
  `Genre` VARCHAR(30),
  `Length` TIME,
  `AverageRating` FLOAT,
  `Synopsis` VARCHAR(1000),
  `Cast` VARCHAR(500),
  `ReviewID` INT NOT NULL,
  FOREIGN KEY(`ReviewID`) REFERENCES `Review`(`ReviewID`),
  PRIMARY KEY(`Title`)
);

```

```

CREATE TABLE `Order` (
  `OrderID` INTEGER NOT NULL UNIQUE,
  `Date` DATE NOT NULL,
  `NoSeniorTickets` INTEGER NOT NULL,
  `NoAdultTickets` INTEGER NOT NULL,
  `NoChildTickets` INTEGER NOT NULL,
  `Time` TIME NOT NULL,
  `TotalCost` DOUBLE NOT NULL,
  `Status` VARCHAR(12) NOT NULL,
  `Cemail` VARCHAR(50) NOT NULL UNIQUE,
  `MovieTitle` VARCHAR(50) NOT NULL,
  `CardNo` INT NOT NULL,
  FOREIGN KEY(`Cemail`) REFERENCES `Customer`(`Email`),
  FOREIGN KEY(`MovieTitle`) REFERENCES `Movie`(`Title`),
  FOREIGN KEY(`Cardno`) REFERENCES `PaymentInfo`(`CardNo`),
  PRIMARY KEY(`OrderID`)
);

```

```

CREATE TABLE `PaymentInfo` (
  `CardNo` INT NOT NULL,
  `NameOnCard` VARCHAR(50) NOT NULL,
  `CVV` INT NOT NULL CHECK(0 <= CVV <= 999),
  `ExpirationDate` DATE NOT NULL,
  `Saved` VARCHAR(3) NOT NULL CHECK (Saved == 'Yes' | Saved == 'No'),
  `Cemail` VARCHAR(50) NOT NULL,
  FOREIGN KEY(`Cemail`) REFERENCES `Customer`(`Email`),
  PRIMARY KEY(`CardNo`)
);

```

```

CREATE TABLE `Prefers` (
  `OrderID` INTEGER NOT NULL REFERENCES `Order`(`OrderID`),
  `TheatreID` INTEGER NOT NULL REFERENCES `Theatre`(`TheatreID`)
);

```

```
CREATE TABLE `PlaysAt` (  
    `MovieTitle` VARCHAR(50) NOT NULL,  
    `TheatreID` INTEGER NOT NULL,  
    `ShowTime` TIME,  
    `Playing` VARCHAR(50) NOT NULL,  
    FOREIGN KEY(`MovieTitle`) REFERENCES `Movie`(`Title`),  
    FOREIGN KEY(`TheatreID`) REFERENCES `Theatre`(`TheatreID`)  
);
```