

**Setup document**

**Organizing the “OnFest” event**

Course: ProP

Group: 31

Dimitar Parpulov - 3207358

Diqin Yang - 3324087

Dongdong Ke - 3338207

Phat Tran - 3335968

# Table of Content

1. [**Table of Content**](#_rcwjo5vji85v)
2. [**Preface**](#_qgfmkdz5c64m)
3. [**Revision History**](#_ifiz6a4xh190)
4. [**Agreements - made with the client**](#_xzm9txt17zs9)
5. [**Process**](#_huoez8clm4si)
   1. [Use cases:](#_us5u4lazfw4)
   2. [Enquiring](#_6ly4q71bn8q2)
   3. [Enrollment](#_k2y2o8t7exke)
   4. [Check-In](#_qfsyn8g111qh)
   5. [Buying and loaning items](#_2mw8fw9qhmof)
   6. [Check-Out](#_9jbdx1m71amy)
6. [**Functional Requirements**](#_4htpfsnx3bya)
7. [**GUI of the applications**](#_g4wzyepsdwf6)
   1. [Application to choose which entrance/exit app to open](#_c7c8r9usz02i)
   2. [Check-In for the festival](#_u4djrrj7dnvp)
   3. [Check-out of the festival](#_boly28wapc1l)
   4. [Camping check-in](#_me22w7o3zsbl)
   5. [Choose which shop or loaning stand you are at](#_z4bl41m3oqr)
   6. [Shop](#_9q8x59p30lbd)
   7. [Choose what food you/ the client would like](#_i6rz1ek7fwv4)
   8. [Choose what drinks you/the client would like](#_v6ifj2h7qmfh)
   9. [Loaning stand](#_p4usasm5bxrg)
   10. [Choose what electronic you/the client would like to loan](#_hq66ixvwr9k3)
8. [**Website Wireframe and Visual Design**](#_sndej2pngn15)
   1. [Home wireframe](#_r40ewsm00mhi)
   2. [Home visual design](#_5j768jnaul91)
9. [**ERD**](#_dmb6894m3h4u)

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Version date | Changes made | Author |
| 1 | 05.03.2018 | Added all sections, filling the information for each section | Dimitar Parpulov, Diqin Yang, Dongdong Ke, Phat Tran |
| 2 | 16.03.2018 | Updated every single section, filled up the use cases, created a Preface | Dimitar Parpulov |
| 3 | 20.03.2018 | Added color to all sections, made it more readable, little updates to all sections | Dimitar Parpulov |
| 4 | 27.03.2018 | Fixed little mistakes, updated extra features |  |

Changes that we made:

# **Preface**

The document contains details of the agreements between the client - Mr. Mikaeil and the team of Sypnosis for the creation of the software for the event OnFest.

In this setup document we will talk through the deliverables as described in the project plan. We will describe the processes, functionalities and user interfaces for every application Sypnosis will build.

# **Agreements - made with the client**

We agreed to deliver the following software for the event organised by our client - Mr. Mikaeil

**Applications:**

- An application to be used at the entrance of the event

- An application to be used at the entrance of the campsite

- An application to be used at the shops

- An application to be used at the loaning stands

- An application to be used when a visitor leaves the event

- An application to be used when a visitor leaves the campsite

- An application for the organization to inspect the status of the event (dashboard)

**Others:**

- A website

- A database

- A presentation about the project

**Extra features that can be added:**

* Newsletter
* A refund option at the shops/loaning stands
* Keep track how much of each item has been sold
* Show the most sold items in the overview app
* Keep track of which employee is at work
* Hire employees
* Employees login with bracelets and automatically opens the GUI of their workplace

# **Process**

### **Use cases:**

Preconditions:

The following preconditions are valid for every use-case mentioned under:

1. The application is running.

2. The user is logged in with their unique username and password.

If there is a precondition unique for an application, it will be mentioned in the specific use-case.

Exceptions:

The following exceptions are valid for every use-case mentioned under:

1. Connection lost

### **Enquiring**

**User task:** Learn more about the event and decide if they want to buy tickets or not

**Exception:** None

**Functional requirements:**

**Success :**

1. People hear about the event
2. They find the website of the event
3. They read all the information they need
4. They decide to buy tickets
5. They continue to enrollment

**They don’t know:**

1. They register for the newsletter (If we add this functionality)

**They decide not to buy tickets:**

1. They leave the website

### **Enrollment**

**User task:** Register and buy tickets and if needed reserve a camping spot

**Exception:** A user wants to buy more than 6 tickets. In this case he/she will have to let one of his/her friends buy the extra(the ones that exceed 6) tickets

**Functional requirements:**

**Success scenario:**

1. A person or a group decides to buy ticket/s
2. Either everybody buys their own ticket or 1 person buys all of the tickets
3. They choose days of access (If we add this functionality)
   1. If less than 3 days -> they receive an email with the pdf of their ticket/s
   2. If the tickets are for 3 days -> they choose if they want to reserve camping
4. The person that bought the tickets specifies how much camping spots he wants (max 6)
5. The same person specifies the number of the tickets of the people that are joining him in the camping spot
6. He/She receives an email with a pdf with the tickets
7. Everybody can now deposit money into their personal accounts
8. Continue to check-in

### **Check-In**

**User task:** Check-in the festival

**Exceptions:** None

**Functional requirements:**

1. User scans the unique QR code

**Success scenario:**

1. QR code gets scanned.
2. System requests information from the database
3. Database returns information about the user
4. Form shows that the entry is successful and if he/she has access to the event
5. Information of the user is stored into the RFID bracelet
6. User enters the festival

**Failure scenario:**

1. No user found in the database
2. Proof of bought ticket required
3. If there is no proof **no access** is granted

### **Buying and loaning items**

**User task:** Buy or loan items

**Exceptions:**

1. Insufficient money into the account of the user. In this case the user is asked to deposit more money to his account and come back when he/she has done that.

2. A user wants a refund (If we add this functionality) then the employee enters a negative purchase in the system

**Functional requirements:**

**Success scenario:**

1. The user chooses items to buy/loan
2. He/she puts the bracelet to the payment console
3. Buying/Loaning is successful

**Failure scenario:**

1. Insufficient funds in the account

**Extra for Loaning:**

* People have to **return all** the items before exiting the festival area for the last time

### **Check-Out**

**User task:** Check-out of the festival

**Exceptions:**

1. Loaned items not returned. In this case the items must be returned before the person is allowed to leave the festival area.
2. There are still money in the bracelet. In this case the money will be automatically returned to the bank account that made the deposit.
3. Somebody forgot something inside after they returned the bracelet. In this case they are not allowed inside of the event without a showing their ticket and they might receive a new bracelet.

**Functional requirements:**

**Success scenario:**

1. Bracelet gets scanned.
2. System requests information from the database
3. Database returns information about the user
4. Form shows if the user has loaned items to return
   1. He/she returns them and receives back the deposit
5. Person is allowed to exit the festival

**Failure scenario:** None

### **Deposit or retrieve money from balance**

**User task:** Deposit or retrieve money

**Exceptions:** Admin deposits money into the account

**Functional requirements:**

**Success scenario:**

1. Visit web page to deposit or retrieve money
2. Enter bank number
3. Enter amount to deposit or retrieve
4. Validate transaction
5. Account balance updated

**Failure scenario:** None

### **NewsLetters**

**User task:** Send the latest information about the event to the users.

**Exceptions:** None

**Functional requirements:**

**Success scenario:**

1.Click the checkbox if the users want to receive the news about the event.

2.Enter e-mail .

3.Send the e-mail about the event to the users’ mailbox.

**Failure scenario:** None

# **Functional Requirements**

**MoSCoW table**

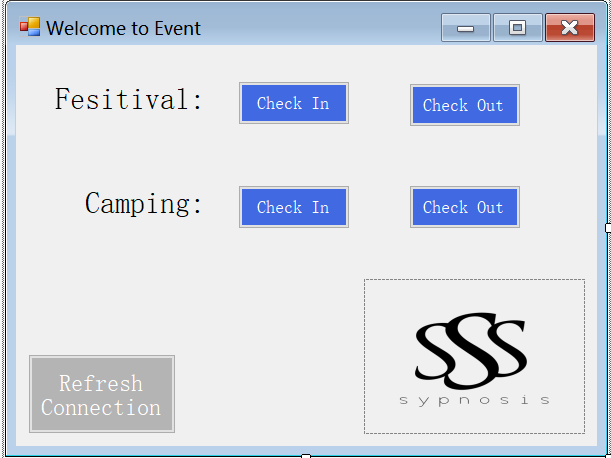
\*Allapplications will be connected to the database

\*All applications will have the option to refresh their connection to the database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirements | Must | Should | Could | Won’t |
| Entrance festival application |  |  |  |  |
| Is entry successful? | x |  |  |  |
| Campsite access? | x |  |  |  |
| User info transferred to RFID bracelet | x |  |  |  |
| Exit festival application |  |  |  |  |
| Is exit successful? | x |  |  |  |
| RFID bracelet returned? |  | x |  |  |
| Loaned items returned? | x |  |  |  |
| Campsite spot is freed? |  |  | x |  |
| Entrance camping site application |  |  |  |  |
| Is entry successful? | x |  |  |  |
| Which camping spot? | x |  |  |  |
| Show roommates info |  |  | x |  |
| Exit camping site application |  |  |  |  |
| Is exit successful? | x |  |  |  |
| Food and Drinks shops application |  |  |  |  |
| Employee check in |  | x |  |  |
| Select multiple foods and drinks | x |  |  |  |
| Remove an item from the order | x |  |  |  |
| Order preview | x |  |  |  |
| Total cost | x |  |  |  |
| Buying of items is successful? | x |  |  |  |
| Decrease the balance of the user | x |  |  |  |
| Loaning stands application |  |  |  |  |
| Employee check in |  | x |  |  |
| Select multiple items | x |  |  |  |
| Remove an item from the order | x |  |  |  |
| Order preview | x |  |  |  |
| Total cost | x |  |  |  |
| Loaning successful? | x |  |  |  |
| Decrease the balance of the user | x |  |  |  |
| Return item | x |  |  |  |
| Overview application |  |  |  |  |
| Access only to Admin users |  | x |  |  |
| Total current visitors | x |  |  |  |
| Bank log | x |  |  |  |
| Overview of shops information | x |  |  |  |
| Update database (delivery) | x |  |  |  |
| Keep track of ”HOT” items |  | x |  |  |
| Website |  |  |  |  |
| Check the information about the Event | x |  |  |  |
| Contact with the Organizer. | x |  |  |  |
| Buy tickets and reserve camping spots. | x |  |  |  |
| Refund a ticket and camping spots. |  |  |  | x |
| Register an account for the website. | x |  |  |  |
| Re-book the tickets and camping spots |  |  |  | x |
| Log in with Facebook or Twitter |  |  | x |  |
| Newsletter |  | x |  |  |
| Admin section |  | x |  |  |
| Add balance to the employees |  | x |  |  |
| Log in with the QR code |  |  | x |  |
| Database |  |  |  |  |
| Holds information about each person | x |  |  |  |
| Holds information about loaned items | x |  |  |  |
| Holds information about purchases made | x |  |  |  |
| Hold information about the tickets bought | x |  |  |  |
| Hold information received from the applications | x |  |  |  |

# **GUI of the applications**

### **Application to choose which entrance/exit app to open**



### **2. Check-In for the festival**

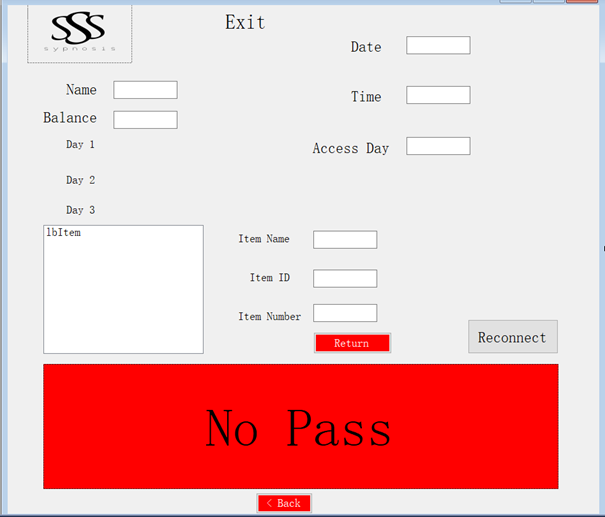
### 

### 

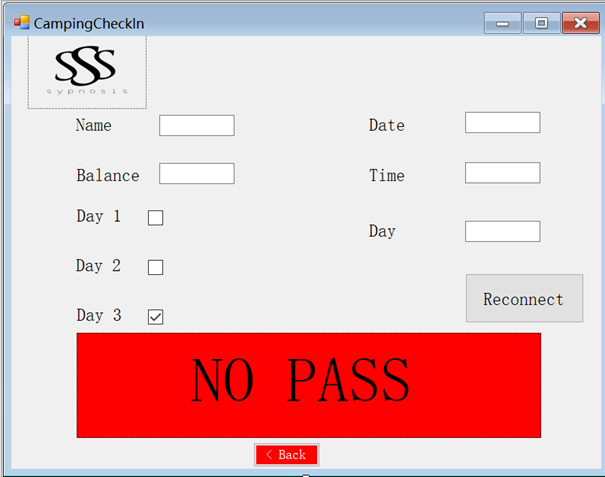
### 

### 

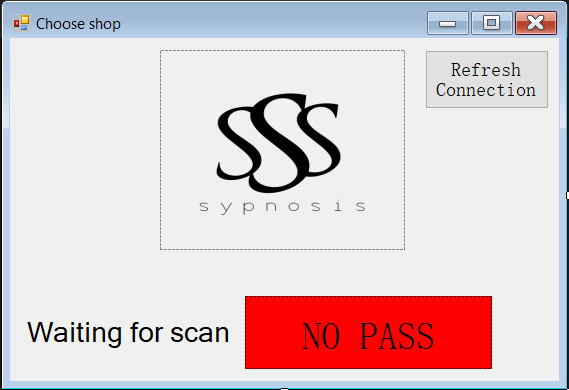
### **3. Check-out of the festival**



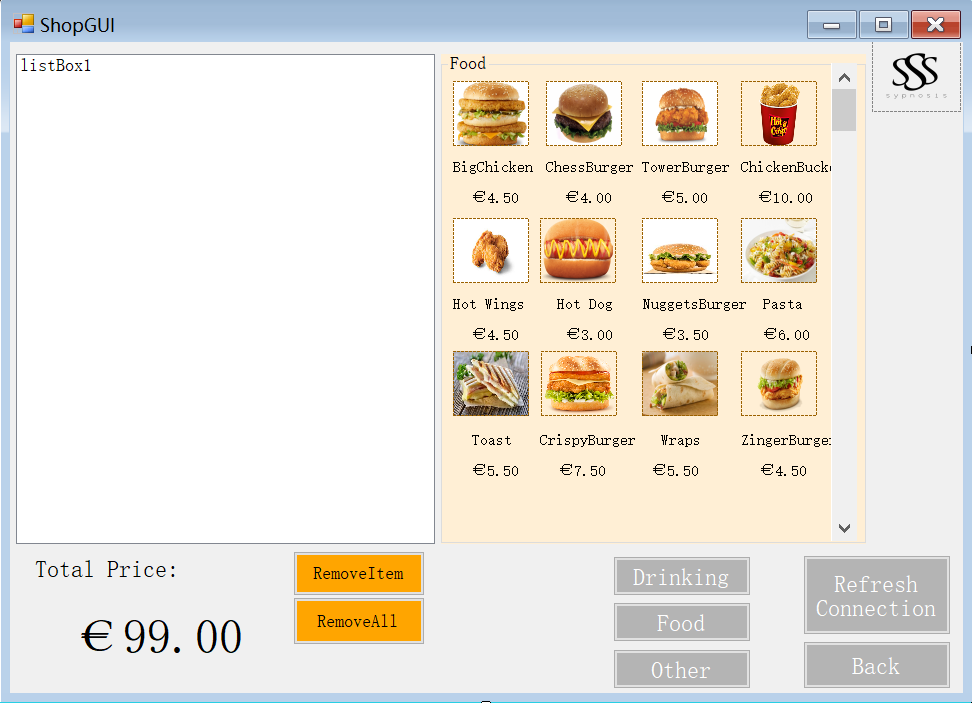
### **4. Camping check-in**



### **5. Employees scans their bracelets**

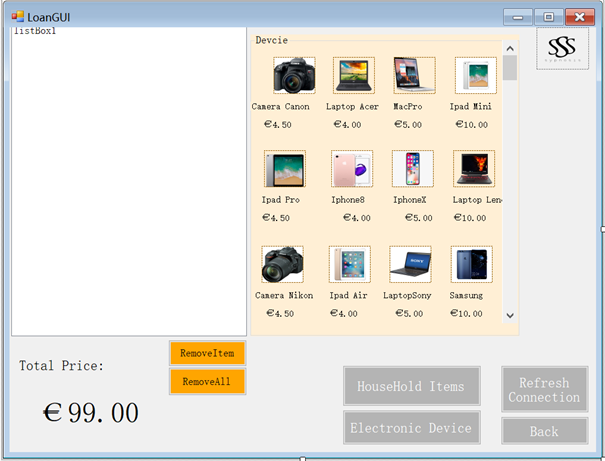


### **6. Shop. Choose what you would like to eat or drink.**

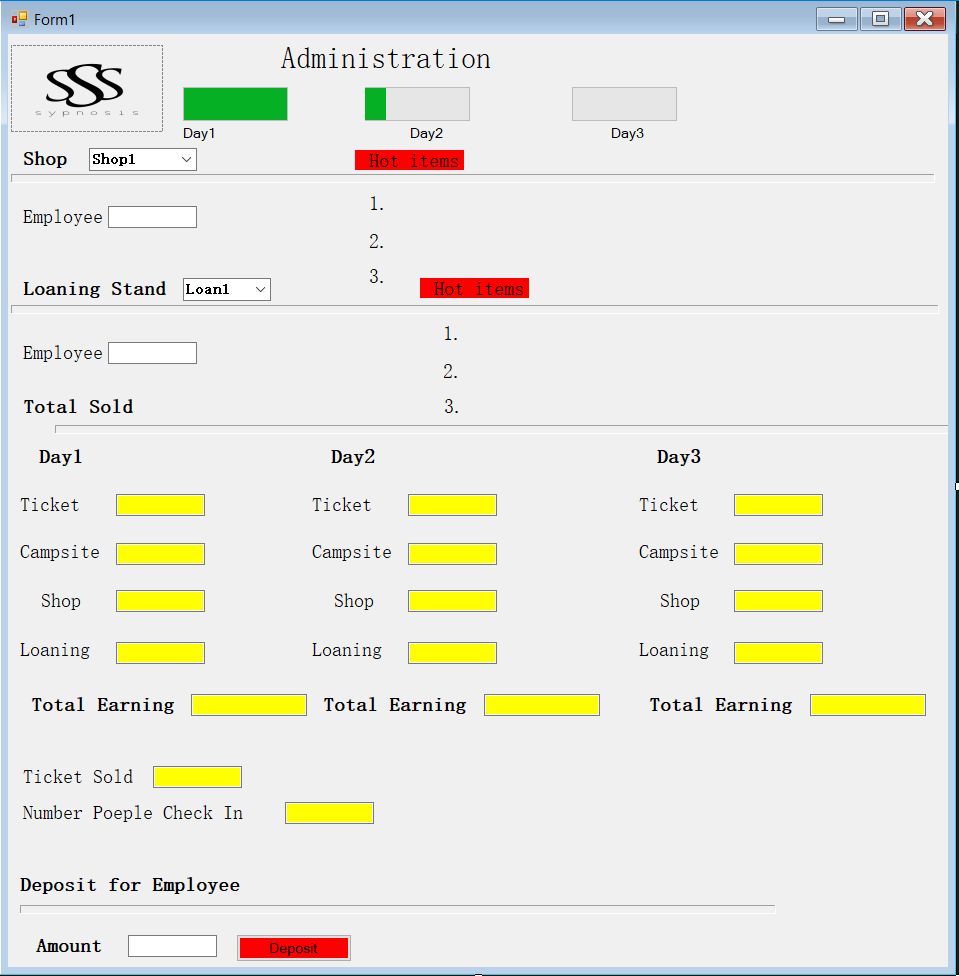


### 

### **7. Loaning stand. Choose what you would like to loan.**



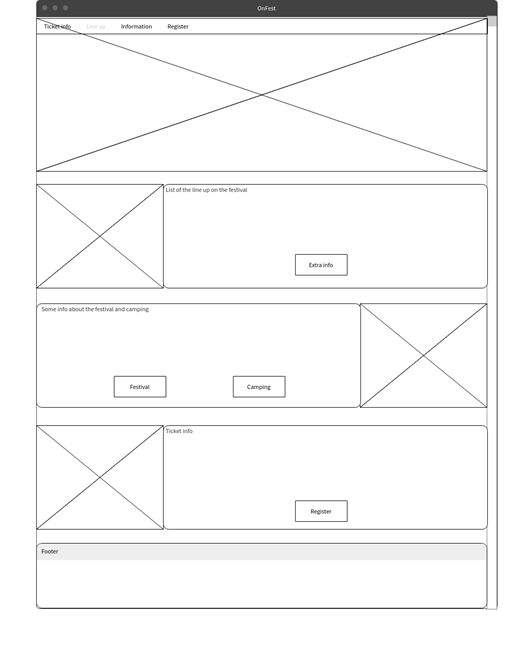
### **7. Administration**



# **Website Wireframe and Visual Design**

The website will be user-friendly. All the information about the event, the onsite camping and everything related to the tickets will be easy to find. People will be able to see the line-up and schedule of the event. Information such as tickets bought, access to the event and the camping area, money spent will be available in the account information of each user.

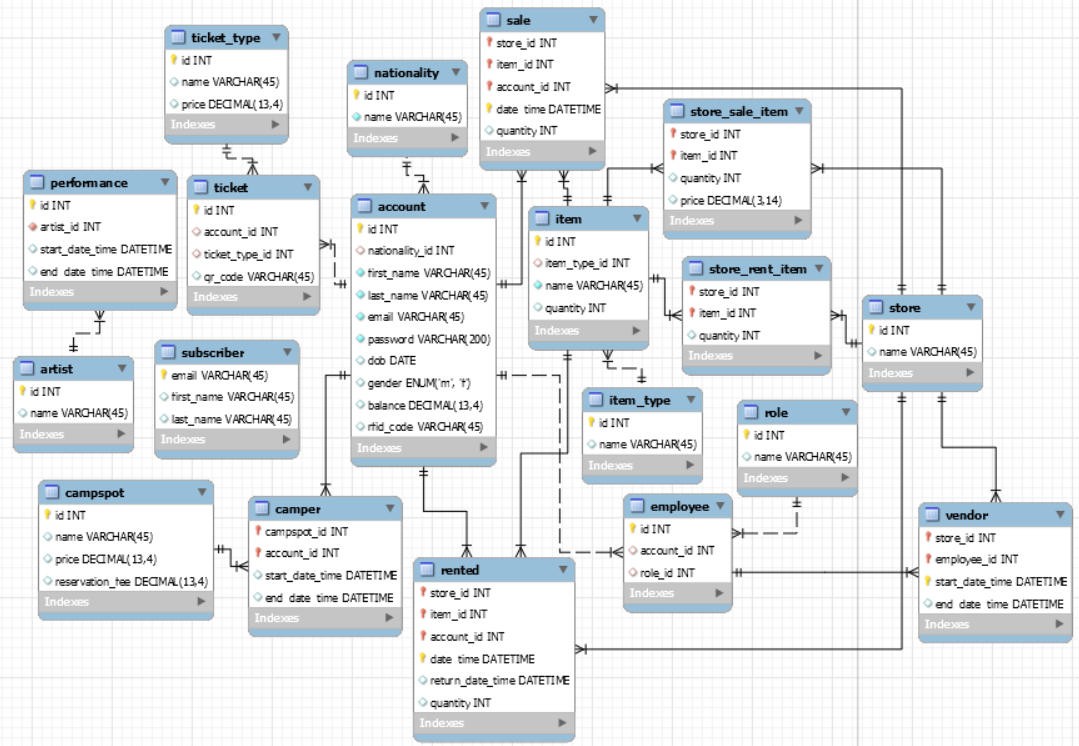
### **Home wireframe**



### **Home visual design**



# **ERD**



**Ticket\_type -** Keeps the type of the ticket(1 day, 2 days, whole weekend) and the price for its type.

**Ticket -** Keeps the information of a ticket of a certain account and a QR code. Each visitor has exactly one ticket.

**Account -** Stores information about visitors of the event. Each visitor must have exactly one Ticket.

**Subscriber** **-** Stores emails of people who want to receive updates on the event.

**Employee** **-** Stores information about each employee.

**Item -** Stores an information about all the items we will have.

**Item\_type -** Stores the types of items we will have.

**Camper** **-** Keeps information for each employee who has access to the camping site.

**Rented -** Stores information about stored items.

**Sale -** Keeps information about which store has sold which item/s