



IS 436 Deliverable 5

Everhart Homies

Mackenzie Harwood, Ren Nguyen, Nathan Chen, Brian Snyder, Sevak Voskerchian



Overview of Project

[Click for more detail](#)

Fun Events System Features

- Store and edit customer information
- Store, edit, and reserve equipment for an event
- Develop event proposals with proper date, time, customer, and equipment information
- Automatically generate event contracts based off of event proposal information
- Electronically sign contracts for events
- Provide storage for contracts, proposals, and other related documents (i.e., tax forms)
- Provide customer and event analytics
- Have events available to view in a calendar
- Accessible to all Fun Events employees when needed



Functional Requirements

- Process oriented
 - The system must allow the management to review and edit information regarding clients, events, employees, schedules, inventory, proposals, vendors, etc.
 - Clients will be able to sign documents electronically.
 - CRM data management will be made available to automatically input customer data.
- Information oriented
 - The system must be able to store information such as the following:
 - Client information such as name, addresses, phone numbers, etc.
 - Employee information such as name, address, phone number, emergency contact, hiredate, etc.
 - Information on the current inventory in the warehouse as well as the inventory that can be rented from vendors
 - Individual items will be able to have detailed information stored.
 - The system will allow access to all documents used for business
 - These include contracts, proposals, tax documents, etc.

Use Scenarios



Use Scenario: Focused Customer

User wants to schedule an event.

Rectangular Snip

1. User browses event schedule for available dates.
2. User browses the equipment schedule for the event.
3. User completes an event proposal.
4. User submits event proposal.
5. User receives an approved event proposal.
6. User receives a contract.
7. User signs contract and confirms event.

Use Scenario: Exploratory Customer

User wants to consider scheduling an event.

1. User reads site marketing material.
2. User browses event schedule for available dates.
3. User browses equipment inventory.
4. User selects multiple equipment.
5. User saves, modifies, or discards an event proposal.
6. User may want to contact a sales representative with questions.

Use Scenario: Employee

Employees need to approve event proposal and create contract.

1. Employee receives an event proposal
2. Employee checks event and equipment schedule for conflicts.
3. Employee approves event proposal.
4. Employee creates a contract with specified information.
5. Employee sends contract to customer.



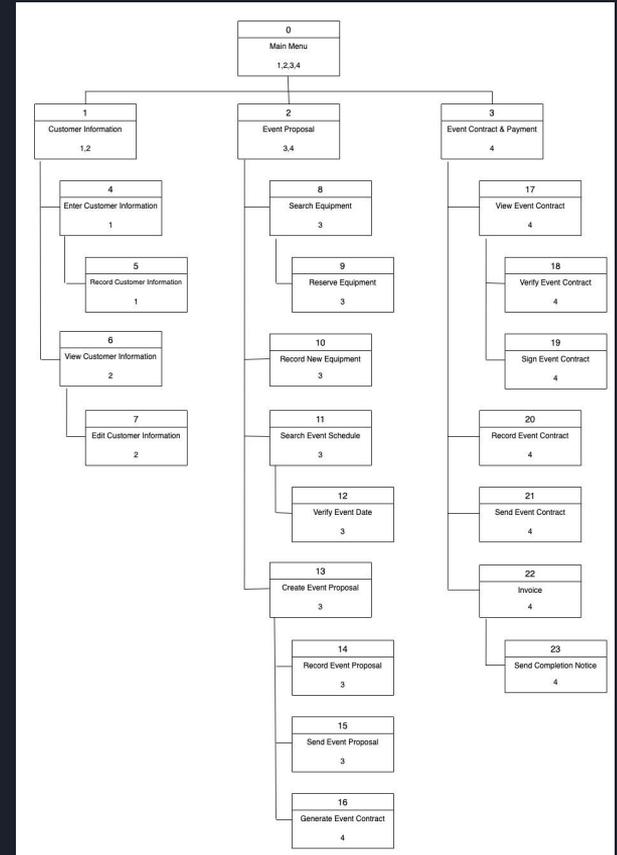
Alternative Chosen

- The three alternative solutions we have considered are Bouncy Castle Network, Goodshuffle Pro, and Inflatable Office
- The alternative solution we have chosen is Goodshuffle Pro mainly due to its price (starting at \$79/month). It can also support multiple users, falling within one of our nonfunctional requirements (supporting 8 users at once)

Interface Structure Diagram

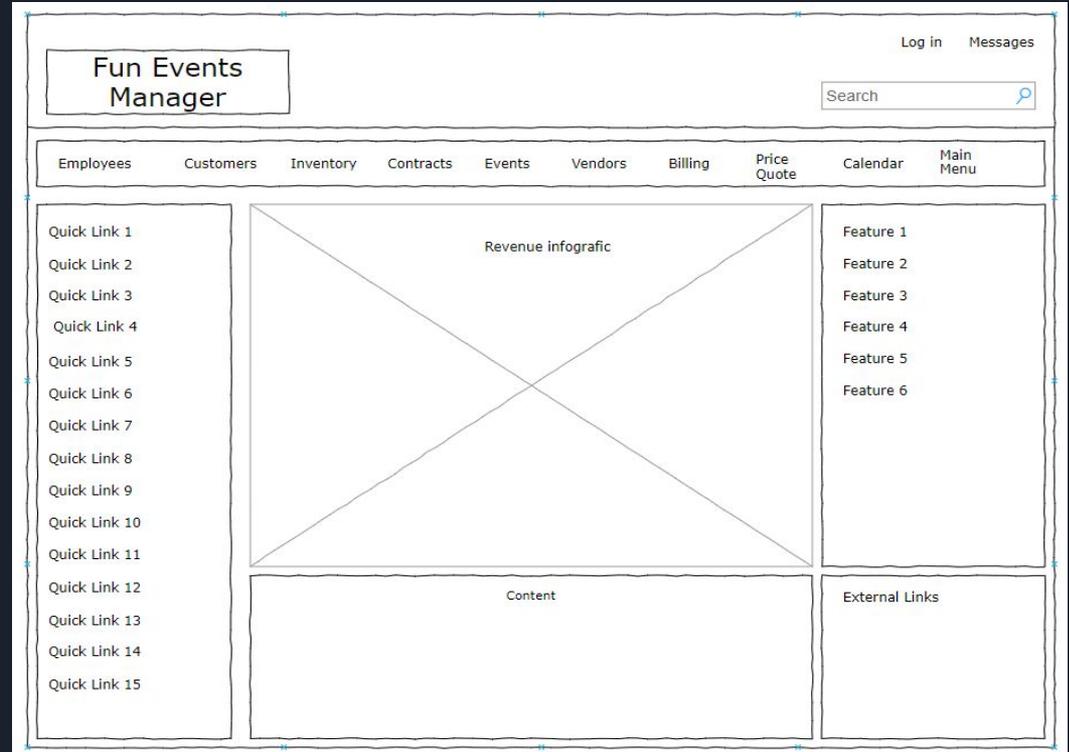
The Interface Structure Diagram (ISD) displays how all screens, forms, proposals, and contracts used by the system are related and how a user navigates from one element to another.

- Corresponding to the Level 0 DFD, there are 4 processes and these processes are listed under 23 unique interfaces
- A user can move to interfaces to:
 - Record new customer information
 - Edit existing customer information
 - Create an event proposal
 - Complete an event contract and payment



Home Page of System

The home page of the system provides easy access to tabs and various links with a customizable quick links feature.



Add Client Page

This page adds a client based off of information.

Basic information was drawn up, however more information would be added based on Fun Event's requests.

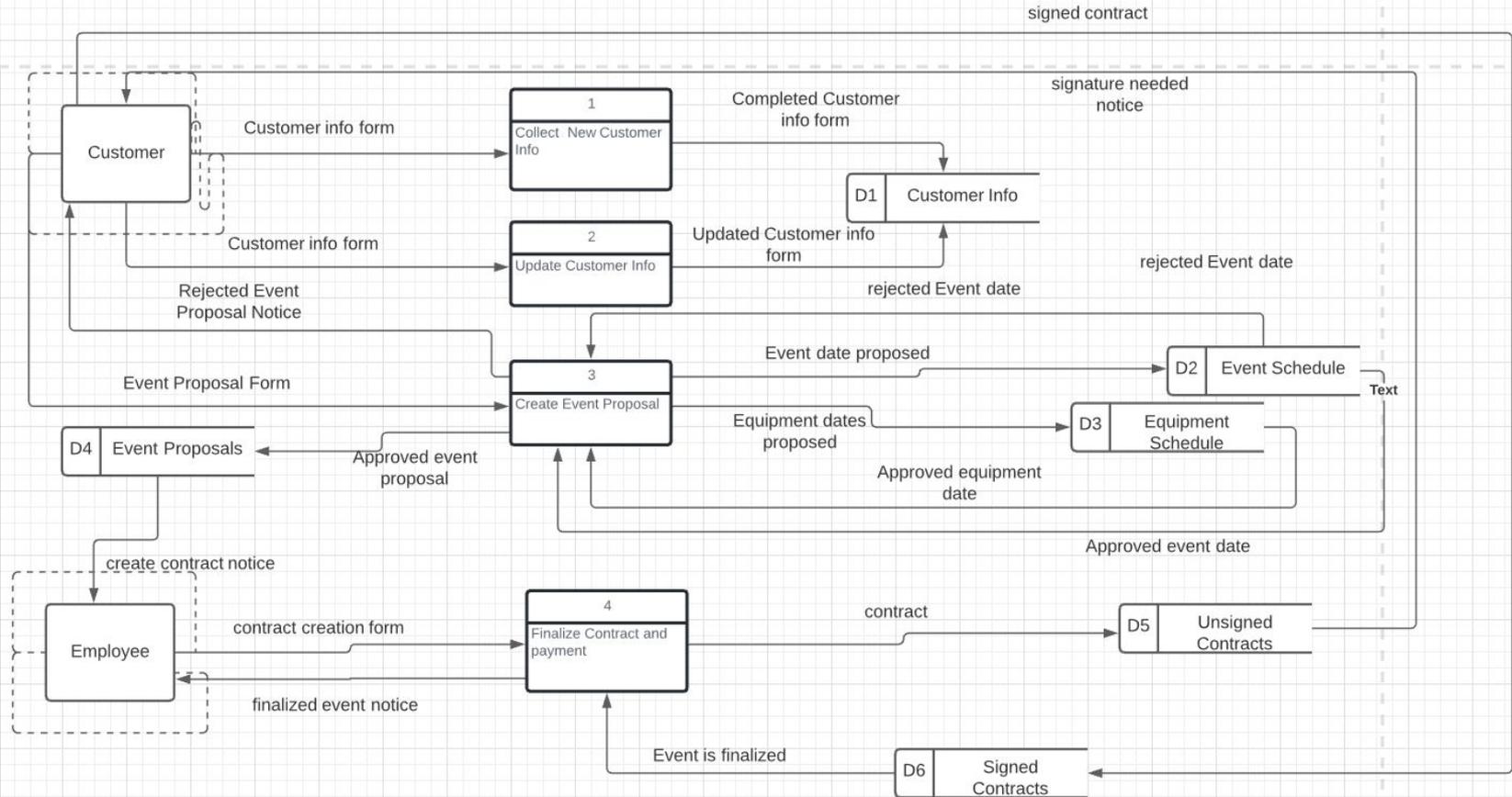
The wireframe shows a web application interface for 'Fun Events Manager'. At the top right, there are links for 'Log in' and 'Messages', and a search bar with a magnifying glass icon. Below this is a horizontal navigation menu with the following items: 'Employees', 'Customers', 'Inventory', 'Contracts', 'Events', 'Vendors', 'Billing', 'Price Quote', 'Calendar', and 'Main Menu'. The main content area contains a form for adding a client with the following fields: 'First name', 'Last Name', 'Phone number', 'Street Address', 'Zip', and 'State' (a dropdown menu). On the right side of the main content area, there is a 'Quick Links' section with a vertical scrollbar and up/down arrow icons.

Client Billing page

This page provides tabs that allow the billing process to be quicker and easier for the employee and client.

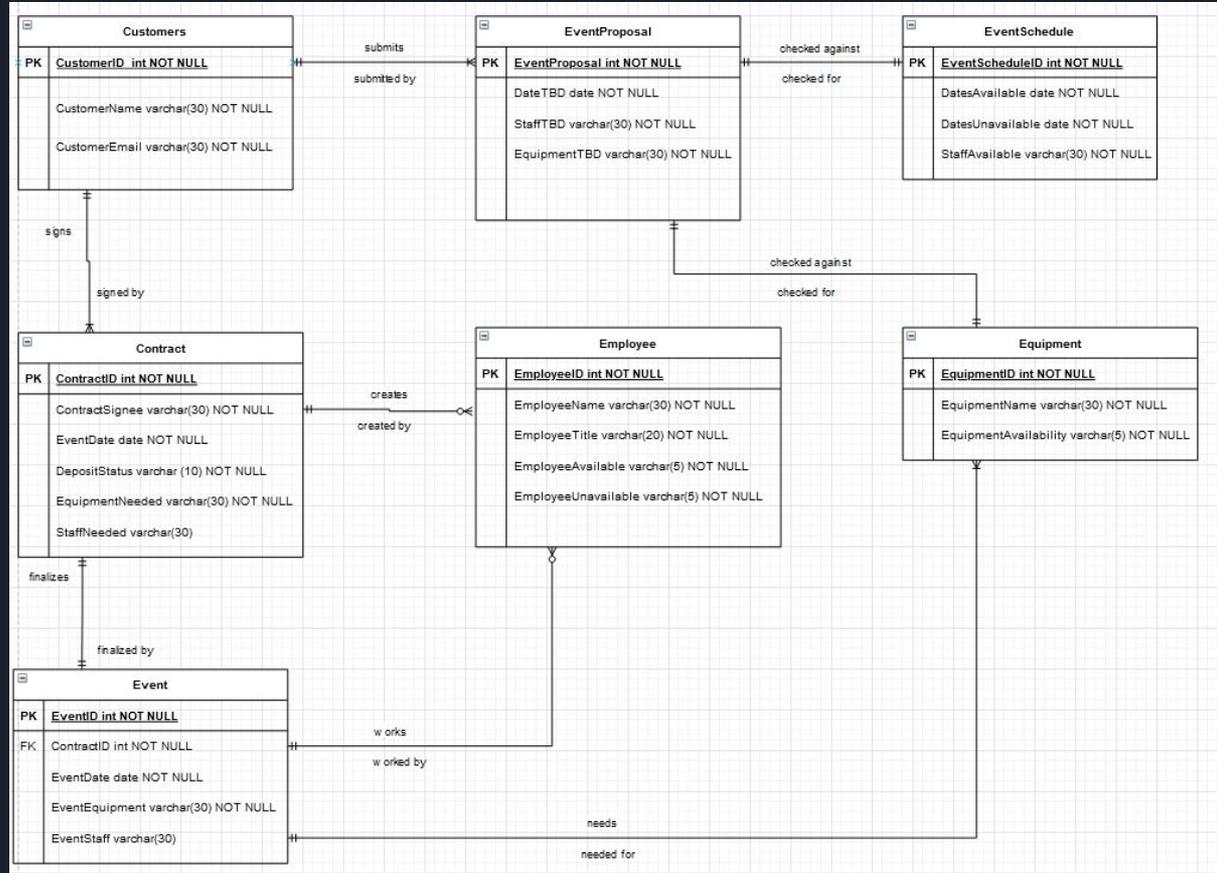
The screenshot displays the 'Fun Events Manager' interface. At the top, there is a search bar and links for 'Log in' and 'Messages'. A navigation bar contains tabs for 'Employees', 'Customers', 'Inventory', 'Contracts', 'Events', 'Vendors', 'Billing', 'Price Quote', 'Calendar', and 'Main Menu'. The main content area is divided into three sections: a 'Quick Link' sidebar on the left with 15 links, a central 'Contract Signature' modal window, and a right sidebar with 'Feature 1' through 'Feature 6'. The modal window shows a signature and a 'Sign Contract' button. Below the modal, a table lists items, including 'Distressed White Farmhouse Cross Back (X-back) Chair' with a quantity of 2, a unit price of \$35.00, and a total of \$70.00. At the bottom, there are placeholders for 'Content' and 'External Links'.

Physical Process Model



Physical Data Model

- PK attributes get “int” data type
- Only one FK
- Attributes related to date get the “date” data type
- All other attributes get “varchar”
- All attributes except “StaffNeeded” and “EventStaff” prohibit null value





Lessons Learned

- We have learned how to communicate with business owners to determine their needs
- We have discovered the complexity of a system for a business, even when the task may appear simple
- Learned the business process of creating models, requirements, and use cases