**SOFTWARE ENGINEERING**

**(IT-314)**

**EVENT PLANNER APPLICATION**

High Level Design Document

(Version 1.0)

**Team no: 14**

**Instructor: Prof. Asim Banerjee**

**Team Mentor: K. Hemantha**

**Date: 29 February, 2016**

**Team Members:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name** | **ID** |
| 1. | Patel Yash | 201301008 |
| 2. | Patel Tej Naginbhai | 201301018 |
| 3. | Anavadiya Yash Kumar | 201301052 |
| 4. | Chauhan Dwimitra Dharmendra | 201301079 |
| 5. | Rahul Shashank Saranjame (Team Leader) | 201301101 |
| 6. | Fatnani Varun Kumarbhai | 201301152 |
| 7. | Nikita Jain | 201301155 |
| 8. | Karri Kiran Satish Reddy | 201301168 |
| 9. | Raman Kumar Douchaniya | 201301202 |

**Version History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version no.** | **Authors** |
| 29/02/2016 | 1.0 | Kiran Reddy, Nikita Jain |

**Table of Contents**

1. Introduction....................................................................................................3

1.1 Purpose.............................................................................................3

1.2 Scope……………............................................................................3

1.3 Overview…......................................................................................3

2. Design Details................................................................................................4

2.1 Main Design Features…..................................................................4

2.2 Use Case Diagrams..........................................................................5

2.3 System Components……………………………………………....6

3. User Interfaces……………………………………………………………..10

4. Module Description.......................................................................................17

4. References………………………………………………………………….18

1. **Introduction** 
   1. **Purpose**

The purpose of High Level Design (HLD) document is to add necessary details to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level. This will help in making development less cost intensive for the cases where the developed product fails to meet the user expectations.

* 1. **Scope**

The High Level Document (HLD) documentation presents the structure of the system, such as the database architecture. The High level design document uses non-technical to half technical terms which should be understandable to the administrators of the system. High level design provides a direction for the coders to proceed with guidelines for a software code structure with minimum number of bugs.

* 1. **Overview**

The overview of the High Level Design will:

* Present all the design aspects and define them in detail.
* Describe the user interface that is being implemented.
* Describe the hardware and software interfaces.
* Describe the performance requirements.
* Include design features and the architecture of the project.
* List of some of the non-functional attributes are:
  + Security
  + Reliability
  + Maintainability
  + Reusability
  + Performance

**Security**

Security is a major issue with respect to the private organisations in a competitive field like event management and advertising. We are issuing a unique username and password which will be provided by the admin to an employee. The employee then logs in with the provided initial username and password to his account to edit his/her details and sets up his/her account. If a user forgets his/her password,a one time password will be sent to the registered email on providing registered username or email. This ensures security of the software.

**Reliability**

A local database will be implemented in which all computers will be connected to the same network. Since no one will interrupt this server, there will be no disturbance. This ensures reliability of the software.

**Maintainability**

Since our software is a web application, it requires very less maintenance. All the users are connected to the same server which is local-host. So it is very easy to update in case of new updates detected.

**Reusability**

The code written and the components used should have the ability to be reused with no problems. Code must be written with well specified comments so that whoever uses the code later can easily get the content and is able to understand the flow and functioning of the program logic. Everything is completely reusable to anyone.

**Performance**

Performance is very important for this project.The updation,deletion and modification to the database must be very fast so that it does not affect the performance of the product and does not give unrealistic time lags to achieve the output.

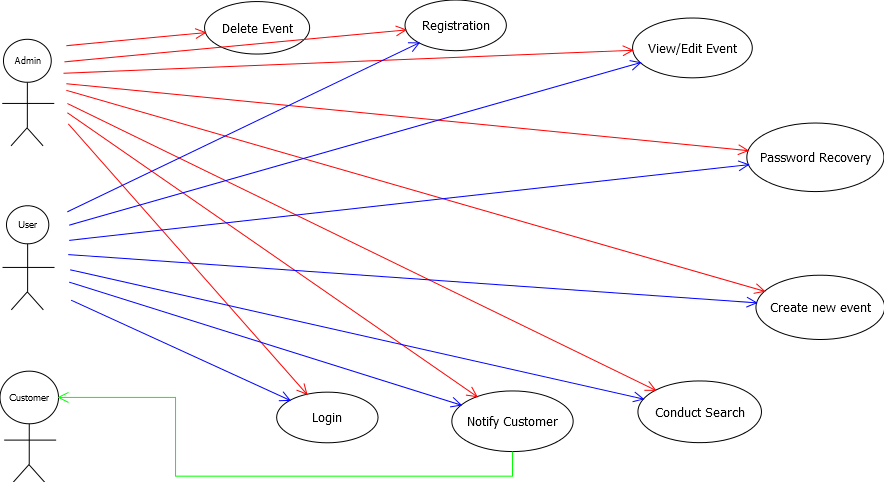
**2. Design Details**

**2.1 Main Design Features**

The main design features include five major parts : the architecture, the user interface design, external interface, the database, process relation and automation. In order to make these designs easier to understand, the design has been illustrated in below diagrams.

**2.2 Use Case Diagrams**

Use case diagrams give us information about the dynamic behaviour of the system. The internal and external agents interacting with the system are known as actors. Every action with the system is started by these actors. Every actor has an association with one or more use cases that give information about the interaction of actors with the system.



**2.3 System Components**

**2.3.1 Database Tables**

* **Event\_ID table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Nullable** | **Unique** |
| Event\_ID | varchar | No | Yes(Primary Key) |
| Name of the event | varchar | No | No |
| Location of the event | varchar | No | No |
| Date of the event | date | No | No |
| Duration of the event | Int | No | No |

* **Costing\_Sheet table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Nullable** | **Unique** |
| Inventory\_ID | varchar | No | Yes(Primary Key) |
| Vendor\_ID | varchar | No | Yes |
| path\_description | varchar | No | Yes |
| Size | int | No | No |
| Quantity | int | No | No |
| status | varchar | No | No |

* **Customer Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Nullable** | **Unique** |
| Customer\_ID | varchar | No | Yes(Primary Key) |
| Name | varchar | No | No |
| phone\_number | varchar | No | No |
| email | varchar | No | No |

* **Conducted For**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Nullable** | **Unique** |
| Event\_ID | varchar | No | Yes(Primary Key) |
| Customer\_ID | varchar | No | Yes(Primary Key) |
| Net\_transaction | long | No | No |

* **Contains**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Nullable** | **Unique** |
| Event\_ID | varchar | No | Yes(Primary Key ) |
| Info\_ID | varchar | No | Yes(Primary Key) |

* **Information**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Nullable** | **Unique** |
| Info\_ID | varchar | No | Yes(Primary Key) |
| Type | varchar | No | No |
| Accepted\_status | varchar | No | No |

* **Payment\_Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Nullable** | **Unique** |
| Info\_ID | varchar | No | Yes(Primary Key) |
| Vendor\_ID | varchar | No | No |

* **Hires**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Nullable** | **Unique** |
| Event\_ID | varchar | No | Yes(Primary Key) |
| Vendor\_ID | varchar | No | Yes(Primary Key) |
| Net\_transaction | long | No | No |

* **Vendor**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Nullable** | **Unique** |
| Vendor\_ID | varchar | No | Yes(Primary Key) |
| Name | varchar | No | No |
| Contact\_Phone | varchar | No | No |
| Address | varchar | No | No |
| Type | varchar | No | No |
| Email\_ID | varchar | No | No |

* **User**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Nullable** | **Unique** |
| User\_ID | varchar | No | Yes(Primary Key) |
| Name | varchar | No | No |
| Date\_of\_birth | date | No | No |
| Password | varchar | No | No |
| Email\_ID | varchar | No | No |
| Designation | varchar | No | No |
| Registered | varchar | No | No |

* **Manages**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Nullable** | **Unique** |
| Event\_ID | varchar | No | Yes(Primary Key) |
| User\_ID | varchar | No | Yes(Primary Key) |

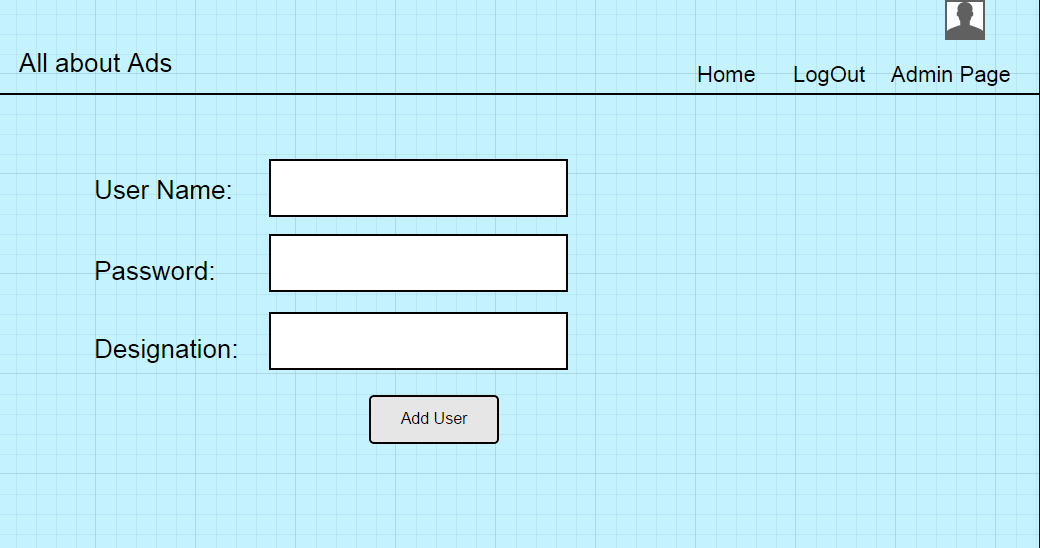
* **Generates**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Nullable** | **Unique** |
| Event\_ID | varchar | No | Yes(Primary Key) |
| Inventory\_ID | varchar | No | Yes(Primary Key) |

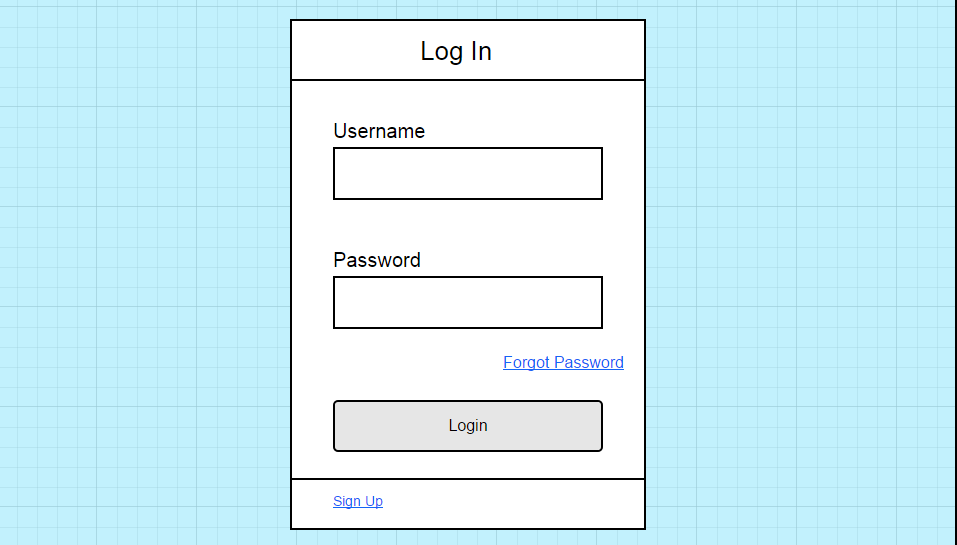
**3. User Interfaces:**

User Interface is a means of communication between a user and a computer program. The system backend provides a frontend for the users to give various commands to the system to get different tasks done. User friendly interfaces have become need for the day when the performance of a software is determined by user frustration. We have designed our user interfaces to be as self explanatory and as user friendly as possible. We have tried to use the laws of human computer interaction to facilitate the interaction of user with the software in a natural and intuitive way. The prototypes of the user interfaces for our software are:

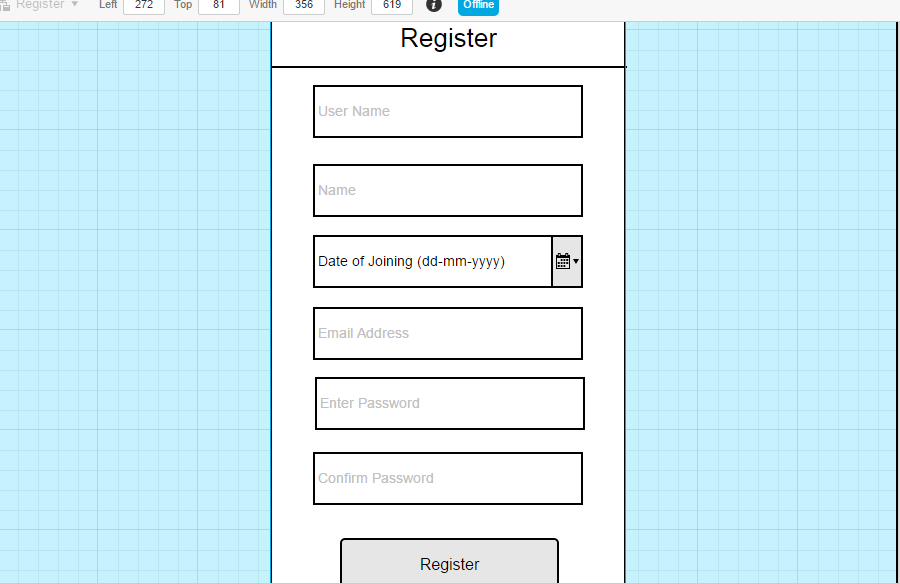
* **Admin Page**

****

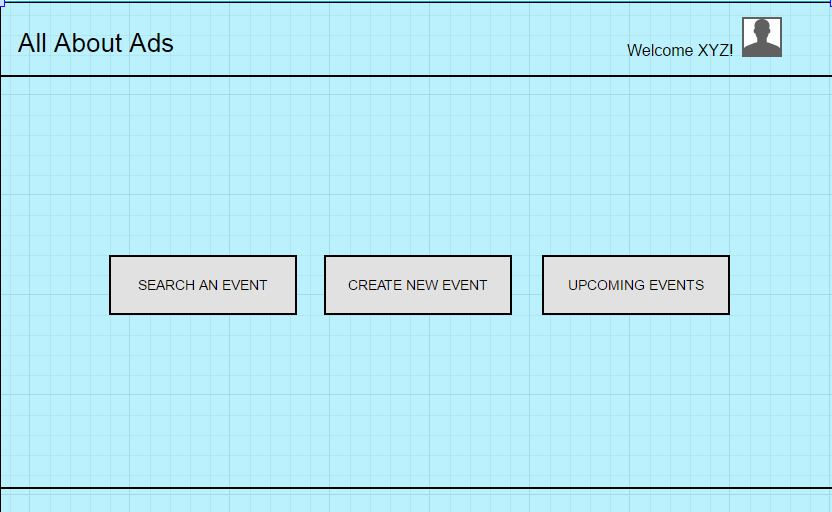
* **Login Page**

****

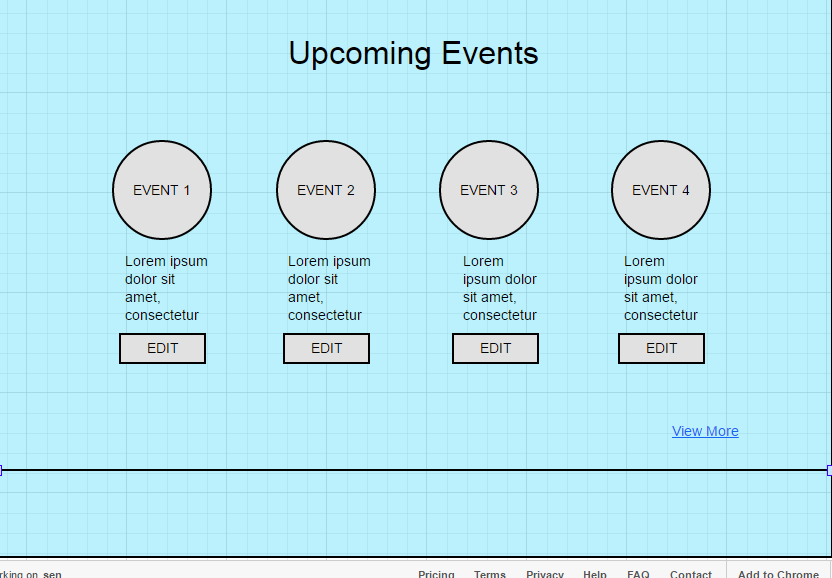
* **Register Page:**

****

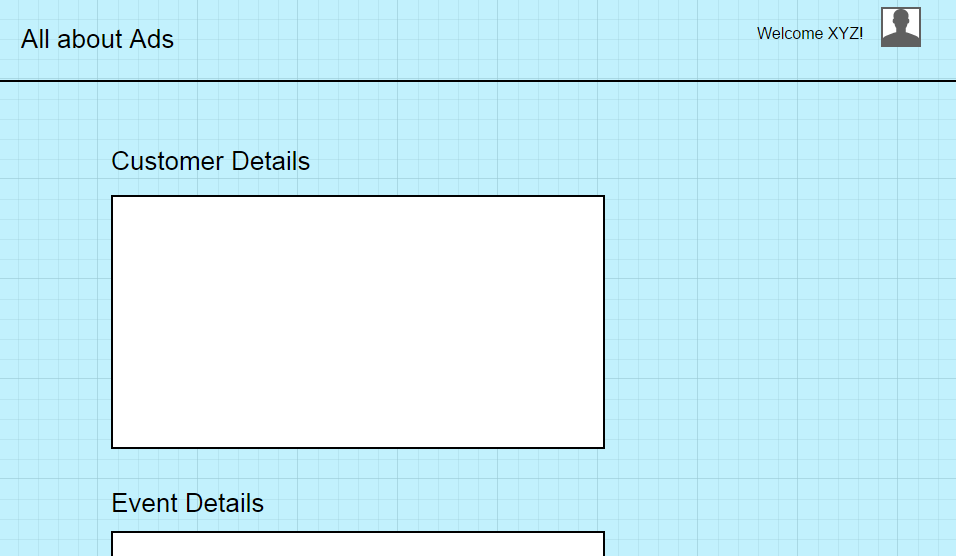
* **Home Page**

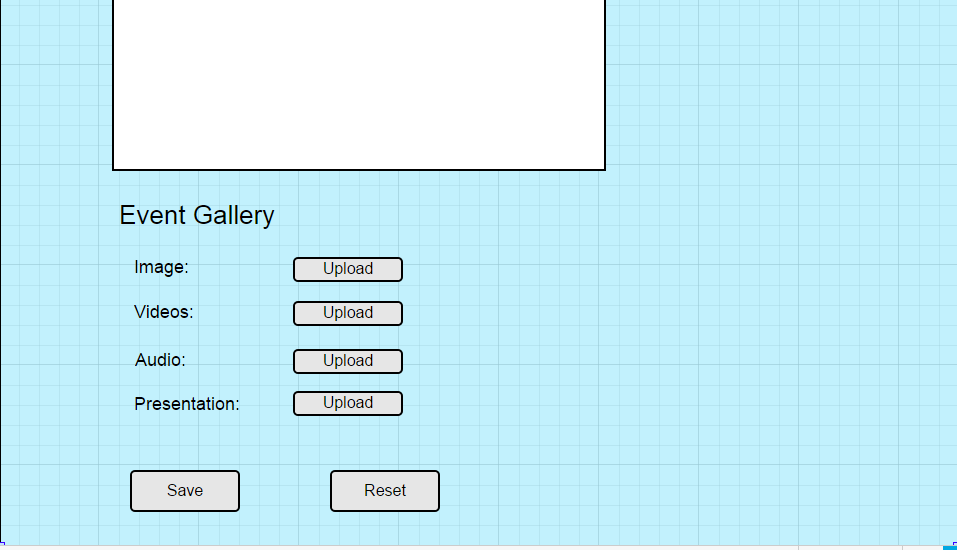
****

* **Upcoming Events:**

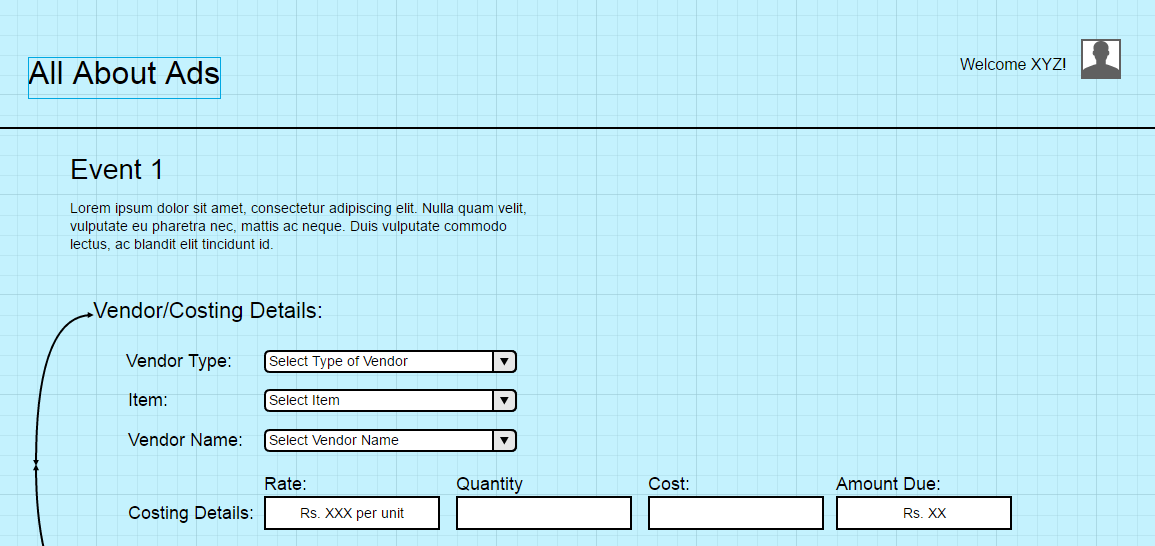
****

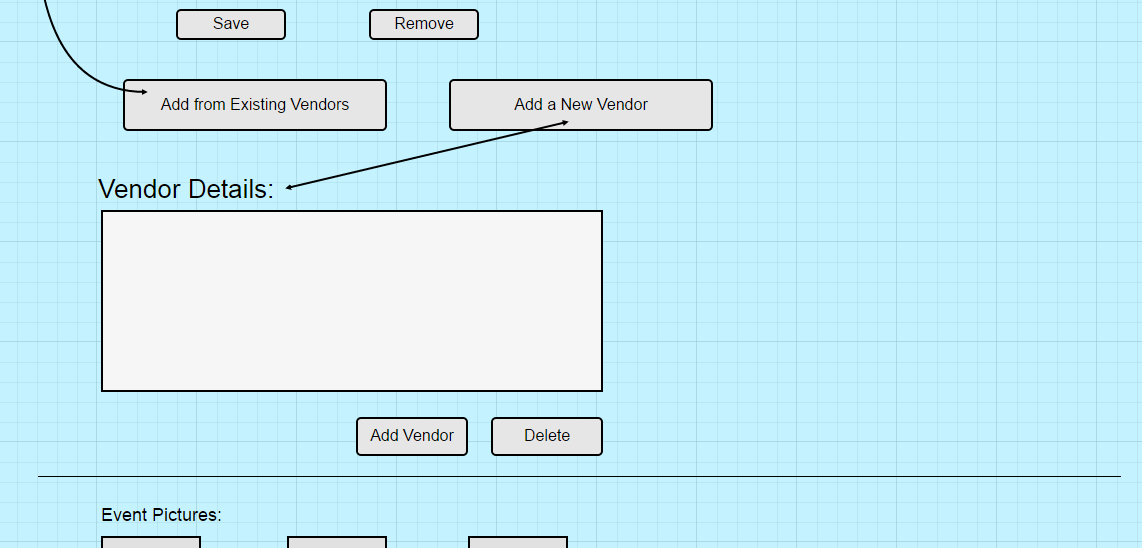
* **Create Event:**

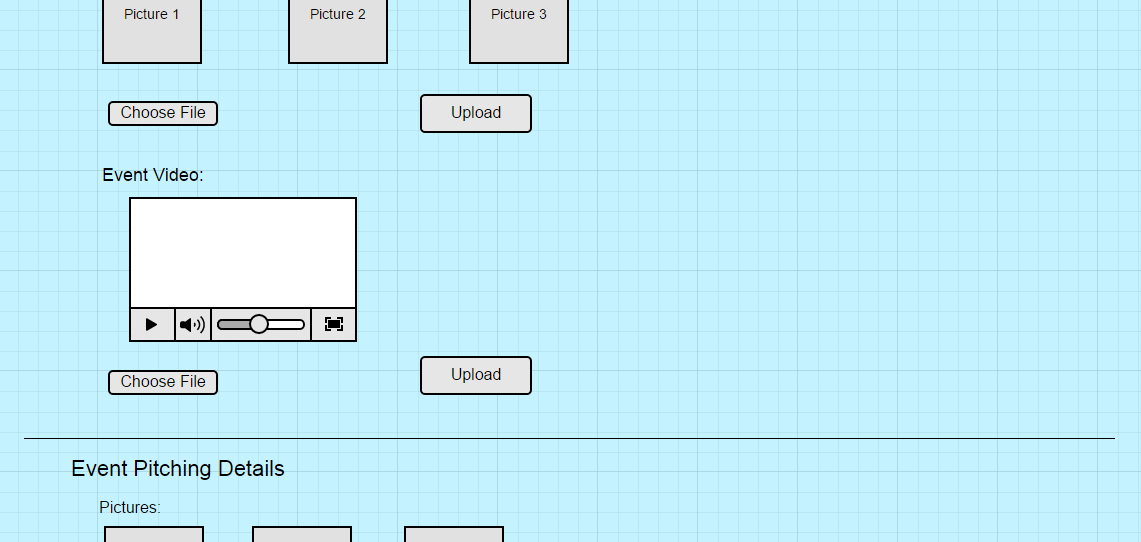
****

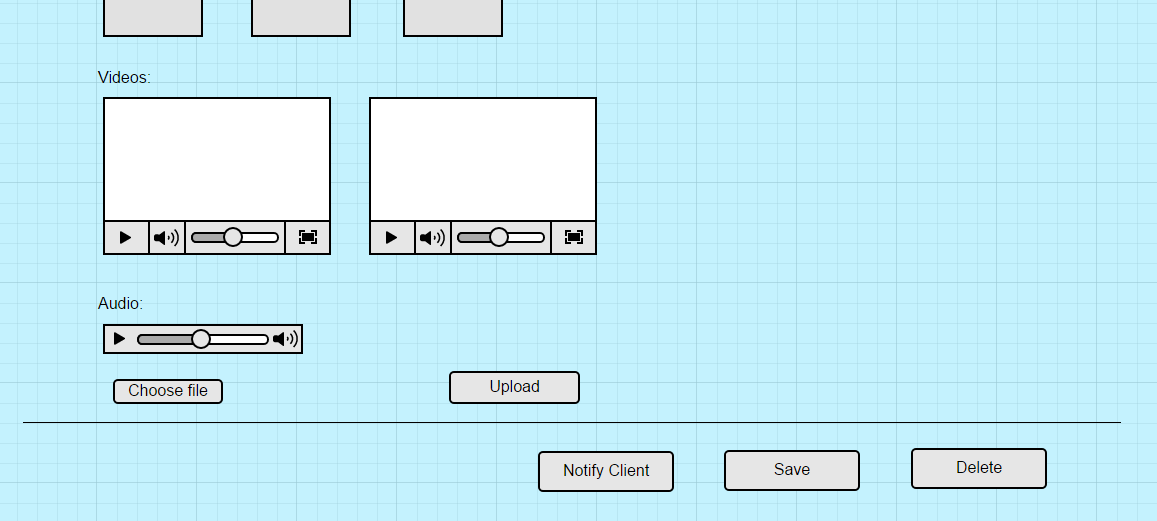
****

* **Event Details:**

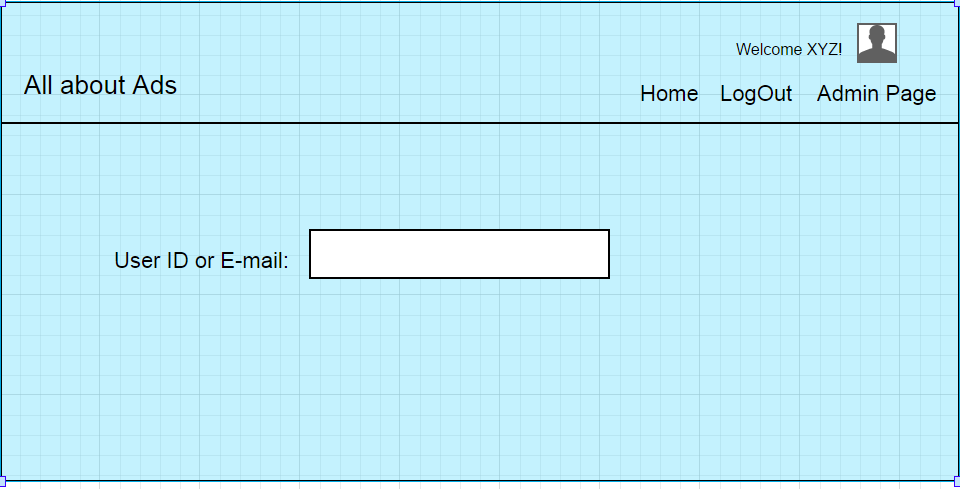
****

****

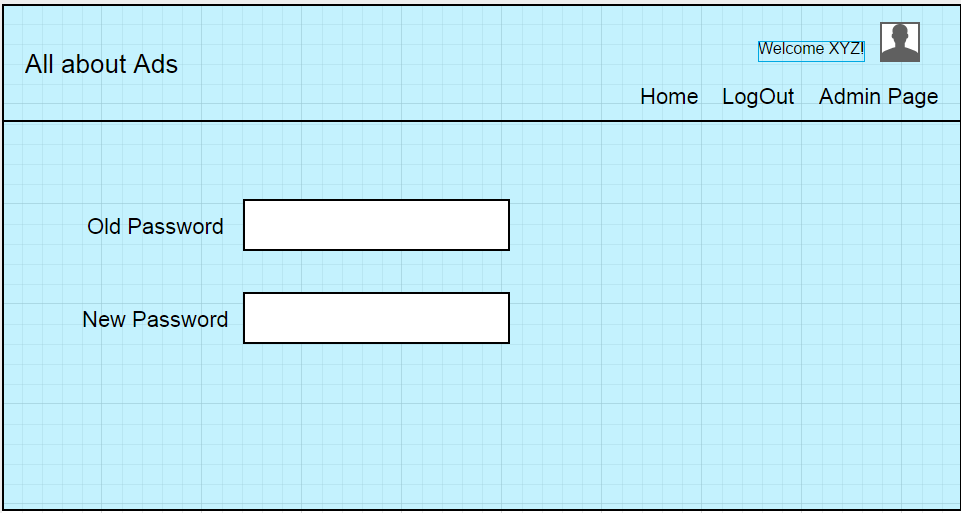
****

****

* **Forgot Password:**

****

* **Change Password:**

****

**4. Module Description**

The different modules present in our software to make it a one big system are:

* **Login Module:**

The login module helps the user to login into application and access all of its functionalities. If the user enters wrong credentials, login failure will occur and the user will be prompted with wrong username or password message.

* **Registration**

The user will be given a username and password by the admin. When the user enters the credentials, he/she will be redirected to registration page where the user fills the details and sets up his/her account . If the user enters wrong credentials he/she will not be registered and will be in the same page.

* **Forgot Password**

On entering the registered username or email id, an email consisting of new random password will be sent to the registered email. Then the user can re login to the application using the sent password. The user can change the password later on if he/she wishes to.

* **Event Search**

A User can search a particular event based on the event name, location, date, budget and event date by entering one or more input.

* **Email-Notification**

An email can be sent to the customer regarding the changes that are going to be done for a particular event in what was planned before. If the customer approves these changes, then only organisation will continue with modifications in the event planning.

* **Create/ Delete/ Modify Event**

An user can create an event at any time. User can modify an existing event at any time by adding or removing the details. The authority of deleting a particular event lies with the admin only. As a part of creating and planning an event, a user can also generate costing sheet for the event.

**5. References:**

1. High level design document v1.0, Team 7, IT314 Software Engineering, Winter 2012-13, DA-IICT
2. Software Requirements Specification, Team 14, IT314 Software Engineering, Winter 2015-16, DA-IICT
3. <https://people.ok.ubc.ca/rlawrenc/research/Students/CJ_05_Design.pdf>
4. https://www.cms.gov/research-statistics-data.../**high**lvltechdesign.**doc**x