LivePortal

System Design

Written by

David Dec
Cole Herzog
Richie Rich

Introduction

LivePortal is a new way to stream content and watch streamed content online. It's primary features are:

- HTML video player that broadcasts streamed video to viewers on the internet
- A chat system for communication between streamers and viewers

Vocabulary

The following terms will be used in the remainder of the document, summarised and defined here for ease of reading:

- Streamer - A user who is using a third-party application to pass video data to our servers to be rendered by the video player

Design Goals

The LivePortal is designed around the idea of creating an online interactive entertainment environment. The system is built to allow multiple streamers stream content from their homes to their own channels and have that content viewed by multiple viewers on the internet.

This online stream is designed to be viewed by general viewers on the internet. This allows users to watch streams, chat with each other, and chat with other users through the chat system.

There are user accounts to facilitate a community. These accounts are how users are logged into chat and how users create their own stream pages.

The stream itself is made with an HTML5 video player. With this design the video player will be lighter-weight and more computationally efficient than competitors.

Key Components

There are multiple components that define the structure of LivePortal. These systems each define an important part of the overall system:

- Application components that are part of creating and editing a character sheet:
    - Web pages
    - Chat engine
    - Video player
- Web service components that are part of managing a user account:
    - Accounts

- Stream pages
- Authentication and security
- Chat login

Web Pages
The web pages are how users will navigate between stream pages, their own account settings pages, and otherwise navigate the site.

Chat Engine
The chat engine is an imbedded IRC chat client. Users will be logged in automatically if they have a user account on the webpage. Stream channel owners and moderators will have the ability to moderate their own chat pages.

Video Player
The video player is the HTML5 player that will be the focus of stream pages. It will receive stream data from a streamer's computer and display it on the webpage for viewers.

Accounts
Accounts are the user accounts that each user will have. This account has a username and password, and also has a follow system to allow users to follow their favorite streamers and be alerted when they are streaming.

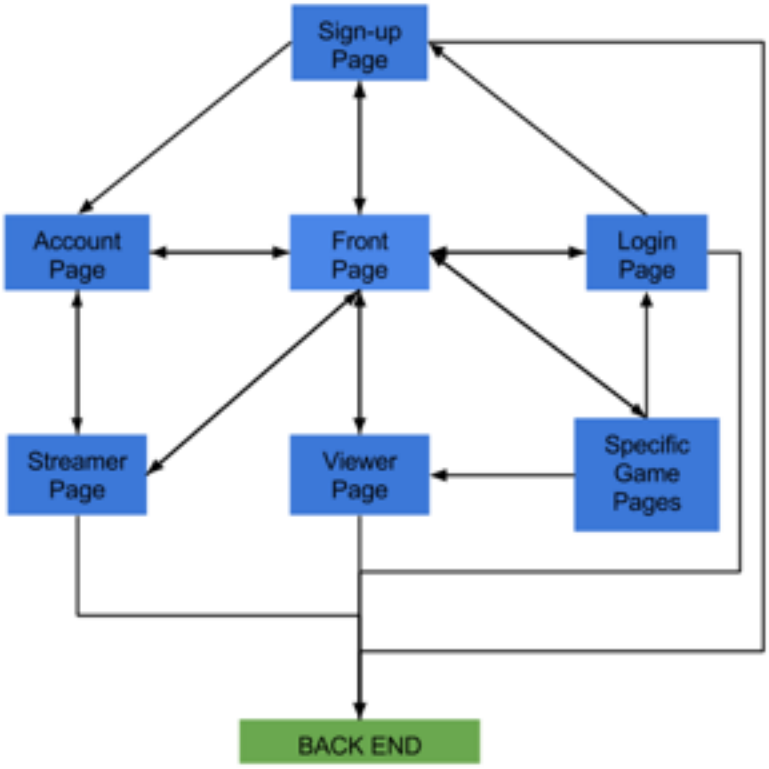
Stream Pages
Stream pages are specific for a certain user. This page holds their HTML5 player streaming video from their broadcast.

The stream page also contains the embedded IRC chat client. Streamers will automatically have administrator permissions for the chat on their own stream page.

Chat Engine
The chat engine is responsible for creating user accounts for users who create accounts as well as logging them into specific chat rooms and assigning permissions to the correct users.

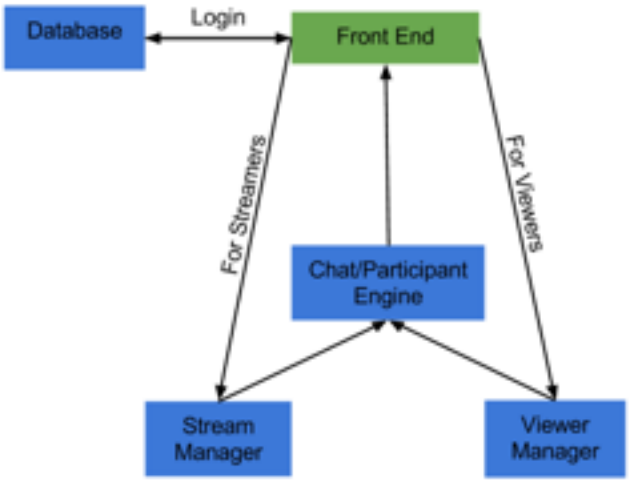
Component Interactions

This is a diagram of the system design interaction:



Front end component

The front end system design is mostly a flow of traffic between pages. Data in the front end is populated by the backend. For example, the specific game pages has to look at the database to find users who are currently streaming and populates what games are displayed based off what users are streaming.



Back end component interaction

The two largest components of the back-end are the stream manager and the view manager. The stream manager handles taking an input from a steamer's PC and takes that data and passes it to the front end for display in the HTML5 video player.

The viewer manager handles data that viewers will see in the front end. It is responsible for passing user information to the chat engine, such as the name of the user so that the chat engine can log in. It also handles user account creation.

The chat engine is an embedded IRC channel. It receives user credentials and information from the viewer and stream manager and allows users to chat using those credentials. Streamers have administrative credentials in the chat engine for their own chat, and they can give privileges to users in their channel, such as giving them moderation privileges or banning them from the chat.

The database is where user credentials and game information is kept. This information includes usernames, passwords, and followed users. Game information is information such as game names and genres. It also contains stream information, such as what game a user is streaming if they are streaming it.