

Stitch

Steam + Twitch

A project by Yikai, Daniel, Nobel, and Harry

Overview:

The website will find and list games that on sale through the steam marketplace using the Steam API. When users click a game on the generated list, they will then be redirected to a webpage displaying the game and it's gameplay found on twitch. Users can also save games and be alerted when the game goes on sale. Layouts and styling will be done using Bootstrap.

Features:

Steam API:

Gathers price info, reviews, and gameplay videos from the Steam API. The Steam API holds a huge variety of information, ranging from featured games to genres of games. We plan on getting most of data from the games specific Steam page, but also plan on getting Steam home page.

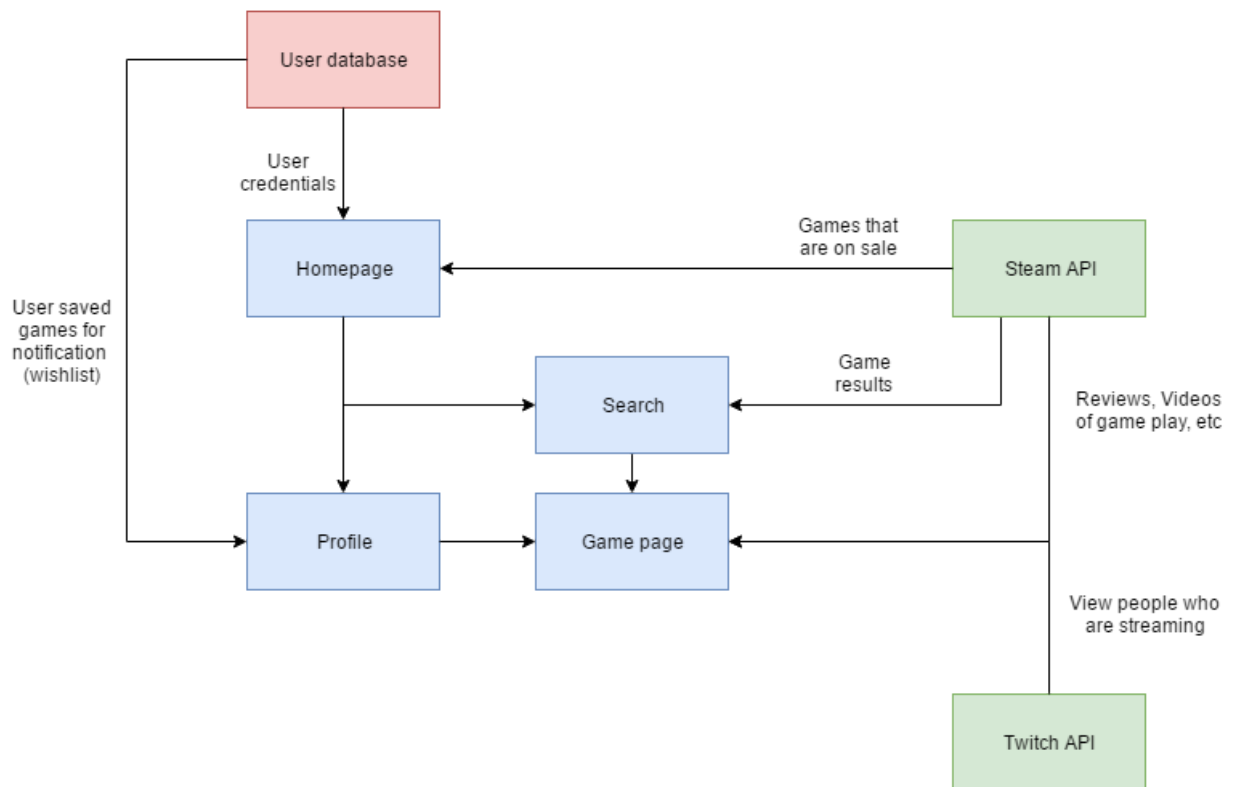
Twitch API:

Many popular games are streamed online via Twitch. Although not every game has a stream up at all times, a majority are. This will allow users to take a peek at the state of the game and the community behind it. What's more, the Twitch chat reveals a new commutative aspect to the games.

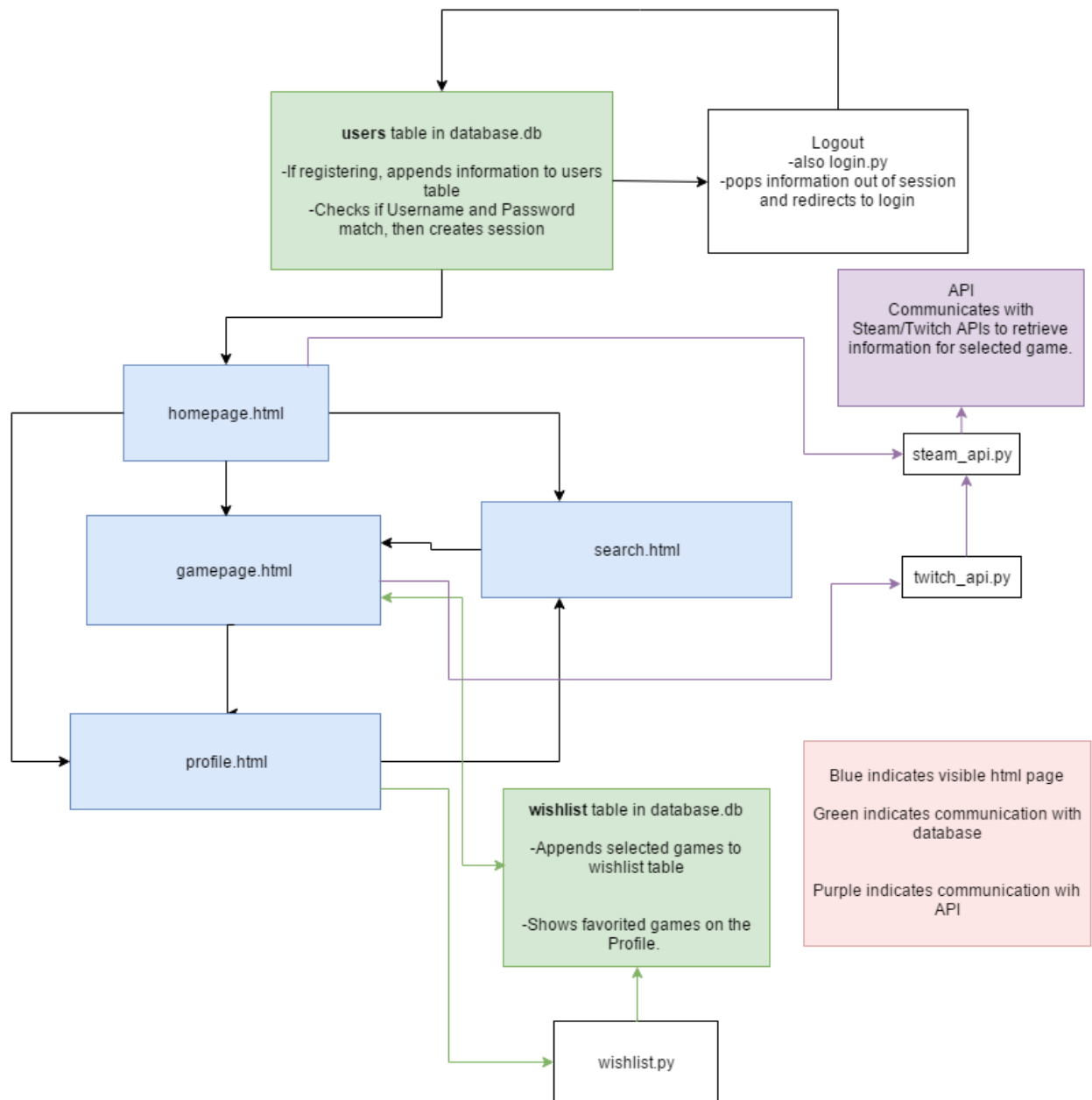
User accounts:

Users can sign up on the website in order to add games to their wishlist and be notified when their desired games are on sale. Additionally, they can link their steam profiles and we can then use the steam API to obtain additional information such as friends list, steam wishlists, and recommended games.

Site Map:



Component Map:



File distribution:

/stitch

app.py

/data

database.db - stores table containing username and password and table containing username and games for wishlist.

app_ids.db - stores table containing game names and corresponding steam ids. Will be used in the search feature.

/utils

login.py - for login and registering.

wishlist.py - for adding a person's game to buy (add to cart feature)

steam_api.py - for using the Steam's api to look up games and reviews. If no games are found, send a message specifying the error that occurred.

twitch_api.py - for streaming a game after a game has been looked up. If no games are found, send a message specifying the error that occurred.

/templates

homepage.html - displays games on sale, and other miscellaneous information
Also includes a dropdown menu to login and register

profile.html - displays user related info such as wish lists, and steam profiles.

gamepage.html - link to page of the selected game with twitch api and steam api information.

search.html - displays search results

/static

any images for backgrounds, buttons, and messages.

css files - We are using bootstrap.

Js files - Javascript for menus

Fonts - includes icons

Database Schema:

database.db

Two tables: users and wishlist

users (to store login info)	
username TEXT	password TEXT
Stores username	Stores password

wishlist (to store profile info)	
username TEXT	games TEXT
Stores username	Stores a wishlisted game

app_ids.db

One table: apps

apps (to store game info)	
name TEXT	id INTEGER
Stores game name	Stores game id

Role Distribution:

Project Manager: Harry

Front end: Yikai - profile and login,

Daniel - homepage and page for game

Back end: Harry + Nobel - api fetching + databases + app.py

Timeline:

12/5/16: Begin designing templates.

12/6/16: Make database tables and finish login.py and register.py. Begin app.py and create website routes

12/7/16: Create homepage and game page. Start steam_api.py and twitch_api.py.

12/8/16: Finish steam_api.py and twitch_api.py

12/9/16: Align apis and html with jinja2 and flask. Finish profile.html

12/10/16 - 12/11/16: Debug website (by trying to break things). Fix any complications

Document Changes:

- We fixed our timeline to better suit our schedule.
- We revised our explanations about our components for clarity, and added cases when the api turns up no results.
- We added a Document Changes section.
- Merged login.html and homepage in a dropdown menu
- Added search.html and search.py to searched items
- Added js and fonts for use in prettifying webpages