

Progress Report - Increment 3 - Group Streamster

1) Team Members

Isaac Wolfe- iww15- IsaacWolfe

Laura Sarmiento - lss16b - Lulairu

Carlos Alfonso - caa16f -
caa16f

Vanessa Myron - vim18 - Vanessa
Myron

Ryan Diaz - rd18d- ryandiaz1090

2) Project Title and Description

Streamsters is a Chrome extension that is made to allow users to add both Twitch and Mixer accounts to view streamers they follow. We support them having the ability to change their game immediately, get a view count, get length of stream, send out custom messages to followers.

3) Accomplishments and overall project status during this increment

In this iteration we focused on finishing the project so all the base functionality was done. We were focused on making sure our application kept the data on closure of the web page as well as preventing duplicate entries. We finalized the styling and completed the theme picker in the settings that allowed us to have a base theme and some alternate users could pick from. Lastly, we added a way to search by top streaming games currently.

4) Challenges, changes in the plan and scope of the project and things that went wrong during this increment

One of the challenges that we faced during the design stage of the project was getting the user icon. In the final product we decided against including this since we were becoming pushed on time and this didn't seem critical.

We did face the challenge of our website needing local storage that we managed to figure out how to use JavaScripts local storage to handle our needs for themes and consistency

once a streamer was added.

Another challenge was having everyone submitting documents at once, even with git, ran into push and pull errors where documents were being worked on at the same time. This issue was resolved as each of us became more familiar with git and making sure we pulled consistency as well as better communication so we weren't doubling up on a specific file.

Due to Covid-19 working remotely in a team has been a challenge for communication as well as keeping track of time schedules. We have started using Zoom to work as a group for big deadlines and discussions and still use GroupMe and Discord although overall remote teamwork has been harder.

5) Team Member Contribution for this increment , *including the sections they wrote or contributed to*

a)

i) *Ryan* - Working on handling permanent storage of following using the google chrome storage api.

ii) *Isaac* - I redid the base theme for the web page to be based off of Twitch and Mixer colors. I also tweaked both the main page and settings page as changes were made and features were scraped, changed, or removed. Lastly, I helped testing and fixing JavaScript functions in the popup.js file and extensions.js file.

iii) *Laura* - Worked on color themes for settings, homepage, and extension. Created the ability to store in local storage the type of theme selected for later use.

iv) *Vanessa* - Worked in the search game features and videos connected to twitch

v) *Carlos* - Contributed to the backend of the project, added ability to sort table headers as well as created the ability to store user stream data for later use.

b) *the requirements and design document, including the sections they wrote or contributed to*

i) *Ryan* - Filled out the basic classes information and functional requirements for the project.

ii) *Isaac* - Created base web design layout document that was later replaced.

iii) *Laura* - I write the Sequence Diagram and help Issac with the web page base design

- iv) Vanessa - Implemented the design interface for the alignment of the games displayed
- v) Carlos- Wrote whatever needed writing when I got to it (basically contributed to this section as needed).

c) the *implementation and testing document*, including the sections they wrote or contributed to

- i) Ryan - Tested APIs using execution-based testing.
- ii) Isaac - Tested the main page for if streamers were added and the page was closed, what happens when adding online/offline streamers, and multiple streamers.
- iii) Laura - Tested theme colors, changed the *jquery.theme* to accommodate to the project, and local storage saving for themes .
- iv) Vanessa - Tested the available games
- v) Carlos- Worked on the execution based non-functional testing section. Tested main functional components in streamers table such as logical statements like handling repeats and offline streamers etc.

d) the *source code* (be detailed about *which* parts of the system each team member contributed to and *how*)

- i) Ryan - Wrote changes to *popup.js* and *background.js* on storage branch
- ii) Isaac - Wrote most of *popup.html*, *extensions.html*, fixed some errors in the JavaScript, helped finalize and stylize the extension files, and wrote all the css files.
- iii) Laura - Wrote changes to *settings.html*, *extension.html* and *popup.html* to make theme changes, all folders and css files in *themeDemo* folder, and changes in *jquery.themes.css* and *jquery.themes.js* for better operation.
- iv) Vanessa - Wrote most of the code for the featured streamers web page
- v) Carlos- Wrote backend support to fix errors we were having in our table. Added the ability to store streamers in *localStorage*. Added the ability to sort through saved streams.

e) the *video or presentation*

- i) Ryan - Walk through the parts I worked on for iteration 2.
- ii) Isaac - Walked through *popup.html*, showed the

extension, and showed off the main functionality of the site

iii) Laura -Walked through jquery.themes.js and background color changes

iv) Vanessa - Walked through the top 50 live games and videos of players playing the games

v) Carlos- Walked through parts for some backend operations. Made and uploaded the video

7) Link to video *Paste here the link to your video*

<https://youtu.be/KbR3FKnpTtE>

R&D

Overview (5 points)

We are proposing a chrome extension to manage your favorite Twitch and Mixer streamers.. Streamster will consolidate your favorite games and streamers into one easy to use button.

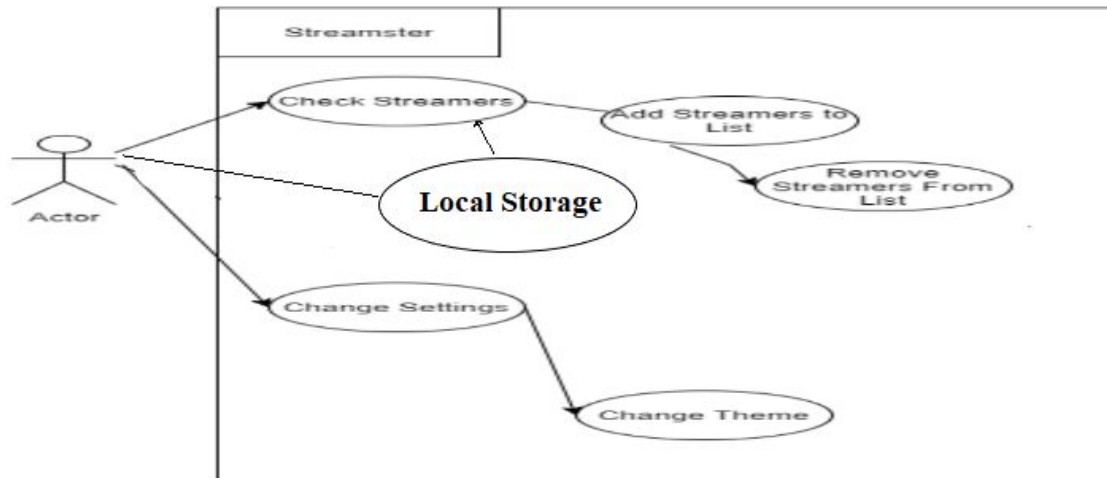
Functional Requirements (10 points)

- 1. Background Themes*
- 2. Setting pane on extensions homepage*
- 3. Clickable window from extension to view current streams with data such as viewers, game, and streamer*
- 4. Custom sort methods of how one wants their list of active streams sorted (such as length of stream, viewer count, game, or streamer name)*
- 5. Display stream information*
- 6. Display the top games and streams currently ongoing.*

Non-functional Requirements (10 points) Perfectly optimized design, the app doesn't require much security besides hiding our own API client key. Webpage

1. Reduce number of API calls through cache evaluation or block get calls
2. Make theming consistent across web pages

Use Case Diagram (10 points)



Class Diagram and/or Sequence Diagrams (15 points) Classes:

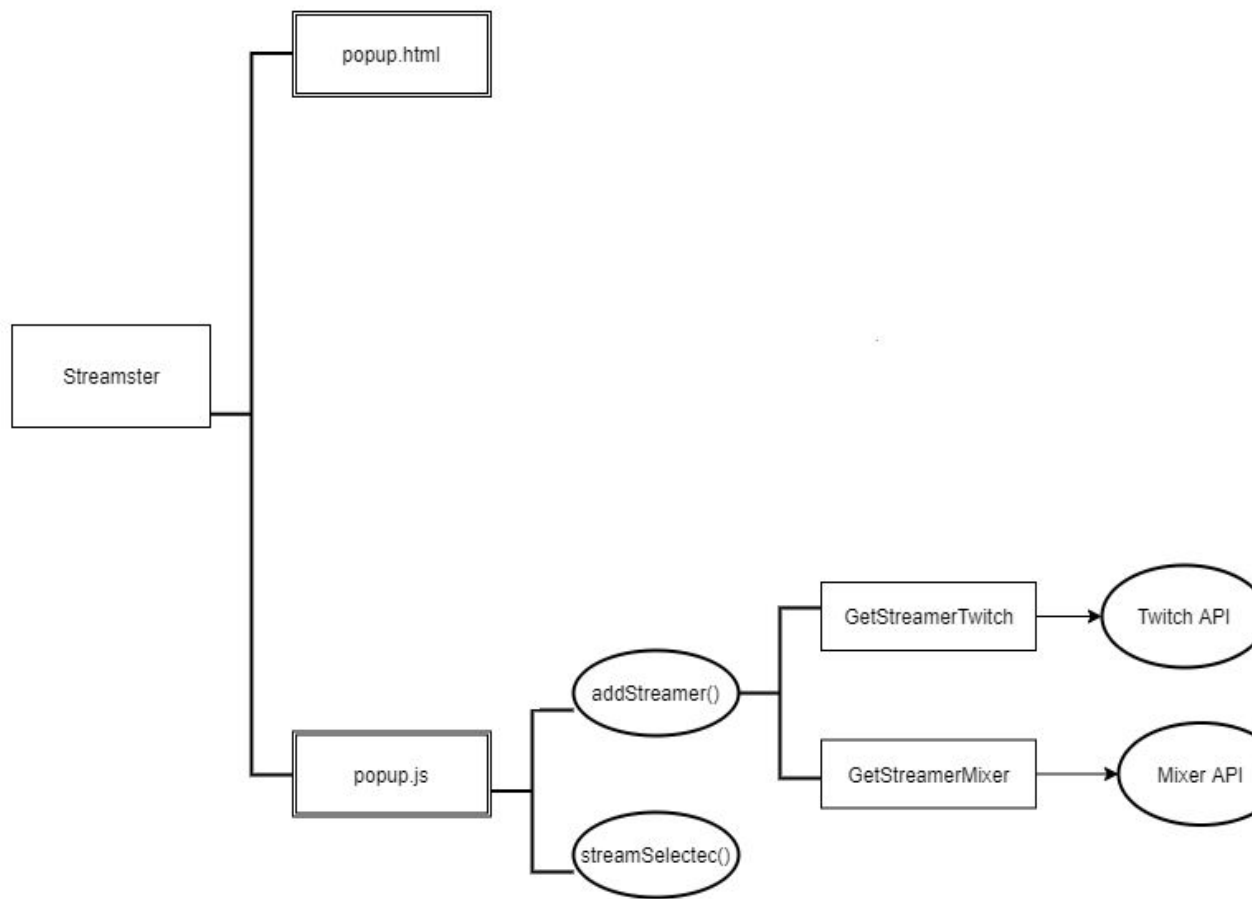
Settings - Handles user settings such as themes, adding their account names to get their follows

Popup - The html page that pops up when the chrome extension icon is clicked. Will show a minified version of the following class

Following - A full webpage that will display more in depth information about who the user is following

Background - Place to store user data either through the chrome.storage api or through our own backend. This will handle continuity to the program.

A **Sequence Diagram** simply depicts *interaction between objects* (or *functions* - in our case - for non-OOP systems) in a sequential order, i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.



Operating Environment (5 points) The software will run within the Google Chrome browser for each user that installs it. All user interaction will be through the Google Chrome browser.

Assumptions and Dependencies (5 points) Assumed factors -> We assume that the API will not change in the future. We also assume that none of the libraries will depreciate. We also assume that our operating system (in this case Google Chrome) can handle all the requirements set forth in our project. We are also using jQuery Themes to handle making themes.

Software Implementation and Testing

Programming Languages (5 points)

List the programming languages use in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may

be).

Javascript, CSS , HTML are our primary languages. We will be using Javascript for the back end and CSS + HTML to handle our web applications. Our reasons for choosing these languages stems from our need to use certain API's like Twitch and Mixer

Platforms, APIs, Databases, and other technologies used (5 points) *List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).*

We will be using the Twitch and Mixer API to make calls so that we can get up to date stream information. We are using local storage to store data as needed

Execution-based Functional Testing (10 points) *Describe how/if you performed functional testing for your project (i.e., tested for the functional requirements listed in your RD).*

We tested API calls to both Twitch and Mixer and made sure that we could deliver on our most important feature to show a streamers name, if they are online, and how many viewers they currently have. We also added the ability for users to search through the top games on twitch.

Execution-based Non-Functional Testing (10 points) *Describe how/if you performed non-functional testing for your project (i.e., tested for the non-functional requirements listed in your RD).*

One way we could have tested the non functional requirements is by gathering data pertaining to the speed at which api calls are done. If there was a way to streamline and increase the speed in which API calls were done that would increase our total API allowed calls and allow for bigger storage.

Non-Execution-based Testing (10 points) *Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).*

We review the code posted on github to check for invalid packaging or other possible errors having to do with how our code fits together. We also demonstrate any new code so that we are all up to date on it. We updated each other on discord when new big updates came out.