

Team LAJICK - Jerry Chang (Project Manager), Issac Kim, Anthony Liang

Abstract : “Life^2” is an app that gamifies habits, to-do’s and goals to keep users on top of their game. Upon registering, users will be given an avatar for which they can nurture, level, and pimp out by keeping in line with their habits and goals. A user can create a list of habits to follow and a set of goals to achieve by creating their own or choosing from presets. By completing these ‘quests’, the user receives bonuses and rewards that they can trade in for items. On the community side, users can check a global leaderboard for their placement. They can also share their custom quests so that other users can use them.

Sidenote : There are a lot of small little details that we’re unsure about like what rewards we can offer and how we can prevent event completion spamming (Currently honor system). These will be worked out over time

Database Schema :

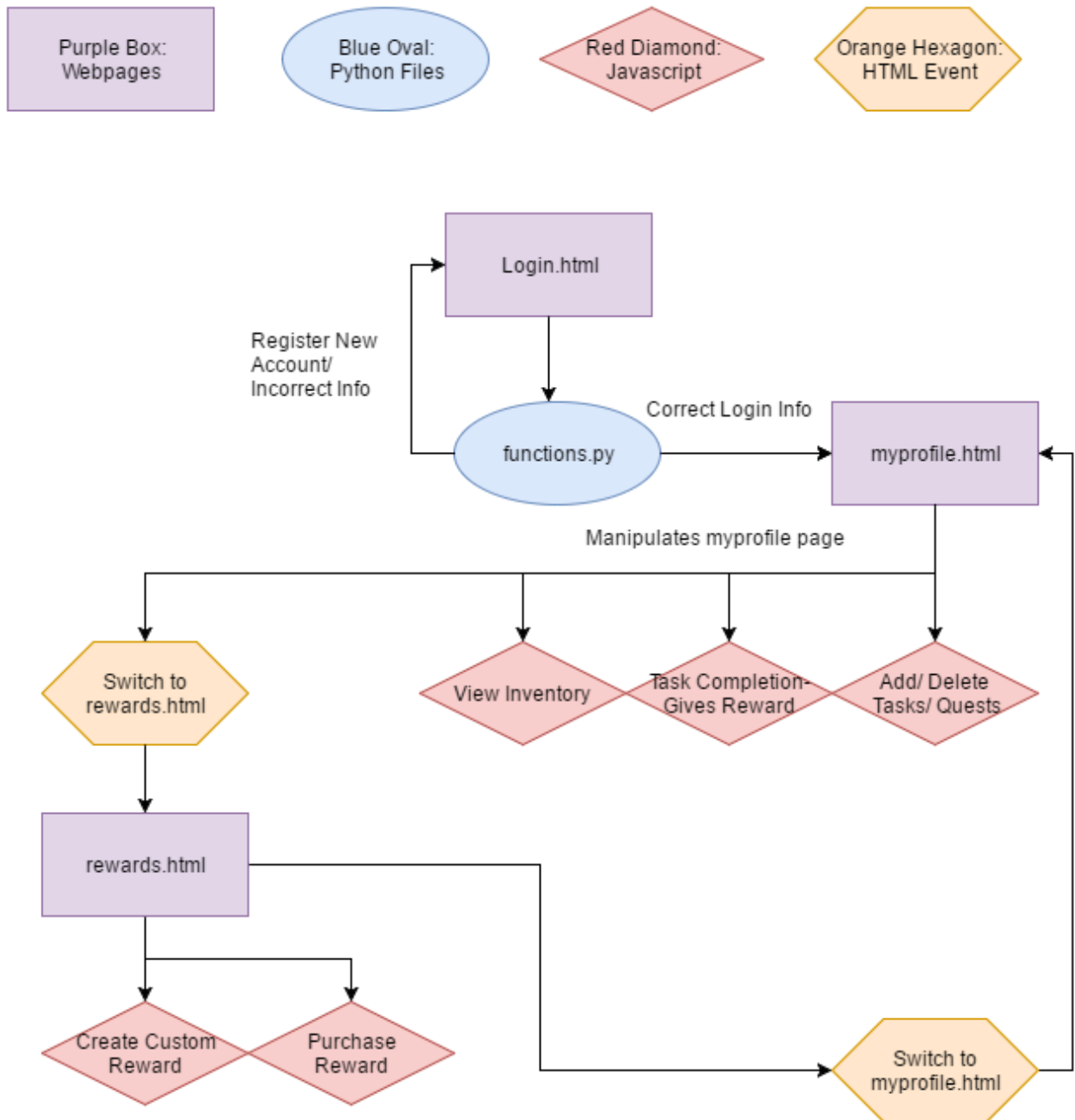
2-3 tables :

- User, hashed password, money, level, current exp
 - Stores profile information
- User, category, name, reward
 - Stores the habits/goals/to-dos
- User, item
 - Stores inventory items?

Timeline:

- 1/9 : Begin frameworks for files
- 1/10 : Get rudimentary front-page working, get rudimentary flask app working, begin function work
- 1/11 : More fleshed out versions of HTML pages and flask app
- 1/12 : Functions (Event addition, Leveling and XP)
- 1/13 : Functions (Event completion, Leveling and XP)
- 1/14 : Functions (Login/Register, Rewards)
- 1/15 : Functions (Inventory, Leaderboards)
- 1/16 : Finish Flask app, begin JS and prettify
- 1/17 : JS and prettify
- 1/18 : JS and prettify
- 1/19 : Miscellaneous things needed
- 1/20 : Get cracking down on bugs
- 1/21 : Finalize overall appearance
- 1/22 : Finish bug testing, minor aesthetic adjustments
- 1/23 : Clean up

Component Map:



Component Description:

Templates (Bootstrap Framework)

-login.html

- css

- heading

- js:

 - division of registration and login

- text form:

 - Refers to route: "/"

 - Reroute to use myprofile.html template with user's login info

-myprofile.html

- css

- heading

- js:

 - task additions

 - task completions

 - current inventory/ gold

- list of user information

 - Displays the amount of gold that the user has

 - Displays the list of habits and tasks that the user created for themselves

-rewards.html

- css

- heading

- js:

 - purchases

 - add custom rewards

- list rewards purchasable by the user

 - gives description of each reward and lists the amount of gold it is worth

Python:

app.py

login [route: "/"] POST

- template: login.html

myprofile [route: "/<user>"] POST

- template: myprofile.html

Description: The myprofile route will render a user dashboard with all the user information (name, level, tasks, todos, dailies, habits) divided by tabs. One of the tabs will also display the rewards user can purchase with their virtual currency.

usersuggestions [route: “/usersuggestions”] POST

- template: userSuggestions.html
- Description: The usersuggestions route will render a collective list of user suggested goals/habits. Every user is able to suggest their own creative and novel goals to inspire other users to take on new goals.

functions.py

getUserInfo(username)

- returns user information in a list (name, level, money, experience)

getUserHabits(username)

- returns the saved tasks, todos, dailies, and habits by the user

getLeaderboardInfo(sortBy)

- returns a list of leaderboard stats depending on requested sort methods (highestLevel, mostGold, mostFinishedTasks)

removeHabits(username, habit)

addHabits(username, habit)

updateInfo(infoType, update)

- takes in the infoType to be updated and updates it

Task Delegation:

Jerry (Project Manager):

- backend, handling the page javascript
- working with databases

Issac:

- frontend, beautifying page and making it more aesthetic
- creating a design for the dashboard
- helping on page javascript and improving user interactivity

Anthony:

- backend
- displaying the global leaderboard, implementing the search functionality
- working with Jerry on functions.py

Site Map:

