

Stardrop Design Doc

Alvin Sze, Kiran Soemardjo, James Sun, Jalen Chen

TARGET SHIP DATE: 2025-12-19

S.T.A.R.D.R.O.P

Space-Time Adventurer **Recharge, Drink, Repair** Omniship Port

Our project aims to simulate a bartending experience using the BoozeAPI to acquire drink recipes (which has a non-alcoholic query filter so we can make a toggle for alcohol) and after customers order, the returned drink information will allow the player to click the appropriate ingredients and drink glass. We thought of a short backstory that places the bar in space, and we will use the WhereTheISSAtAPI to tell the player what city the bar is currently above. Since the bar is in space, we decided customers will pay in gold, which will be converted to USD using the GoldAPI, where we can also use the given historical data to match it with the ISS location at different times (allowing us to perhaps add a time travel function).

COMPONENTS:

`__init__.py` [Author: Jalen + Kiran]:

Contains Flask app. Changes between HTML files. Creates database.

`game_state.py` [Author: Alvin + Kiran + Jalen]:

Contains game information and interacts with the database. Stores all static game information (i.e. NPC dialogue/drink prices). Interacts with API and database to query for data and change HTML.

`recipes.py` [Author: Alvin]:

Stores information on ingredients on hand, how much each ingredient costs, and how ingredients change the color of an image.

`index.html` [Author: James]:

Template for the landing page. Displays button to create account, button to log in and button to start game.

`game_scene.html` [Author: Alvin + James]:

Displays the actual game. Layered images to display graphics and buttons to interact with the game or go back to home. Takes information from

game_state.py to display correct information. Changes database to save account game data.

login.html [Author: James]:

Takes input to log the player in. Checks database.

createaccount.html [Author: Jalen]:

Takes input to create an account and changes the database.

player.db [Author: Kiran + Alvin]:

Stores game information for each account. Each char_x_interact stores how many interactions with each customer (if we have enough time.)

DATABASES:

Players		
TEXT	username	PK Identifier for each player account.
TEXT	name	Name for the bartender.
TEXT	password	Password for account.
FLOAT	money_earned	Money earned from sales.
INTEGER	char_1_interact	How many interactions with NPC 1.
INTEGER	char_2_interact	How many interactions with NPC 2.
INTEGER	char_x_interact	How many interactions with NPC x...
DATETIME	time_period	Date currently time traveled to.
BOOL	alcohol_on	Toggle for alcohol in drinks.

Ingredients		
TEXT	name	PK Identifier for ingredient
TEXT	color	Color to change the image hue. So we don't have to hardcode each drink color, we just take the sum of each ingredient color.
FLOAT	price	Price of individual ingredients. Again to avoid hardcoding, we can just add all ingredient prices for the drink price.

FRONT-END FRAMEWORK:

Foundation: We chose Foundation for its convenient, straightforward documentation and "building blocks" (BB) codes since we plan to use minimal frontend, replacing it with layered images instead. We will however, need many buttons, resizable containers to hold the images, and a switch for the alcohol toggle.

APIs:

BoozeAPI: <https://boozeapi.com/>

Returns drink information which will change the price of the drink and image displayed using the returned ingredients list.

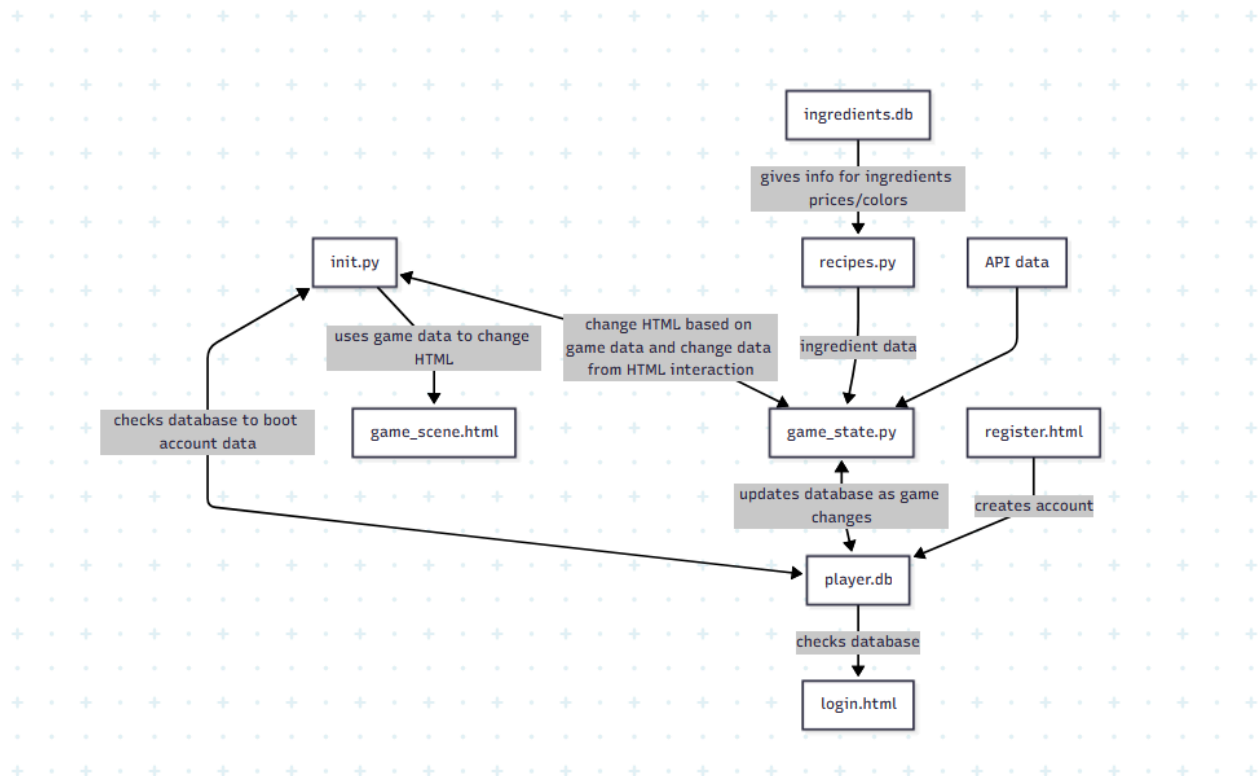
WhereTheISSAtAPI: <https://wheretheiss.at/w/developer>

Returns the city that the bar is above at the time the player time traveled to (essentially replacing the ISS with the bar for simplicity).

GoldAPI: <https://www.goldapi.io/>

Returns how much gold a drink is worth at the time the player time traveled to. Rounding to the nearest ounce, each NPC will pay in one ounce gold coins, which is tracked by amount earned.

COMPONENT MAP:



SITE MAP:

