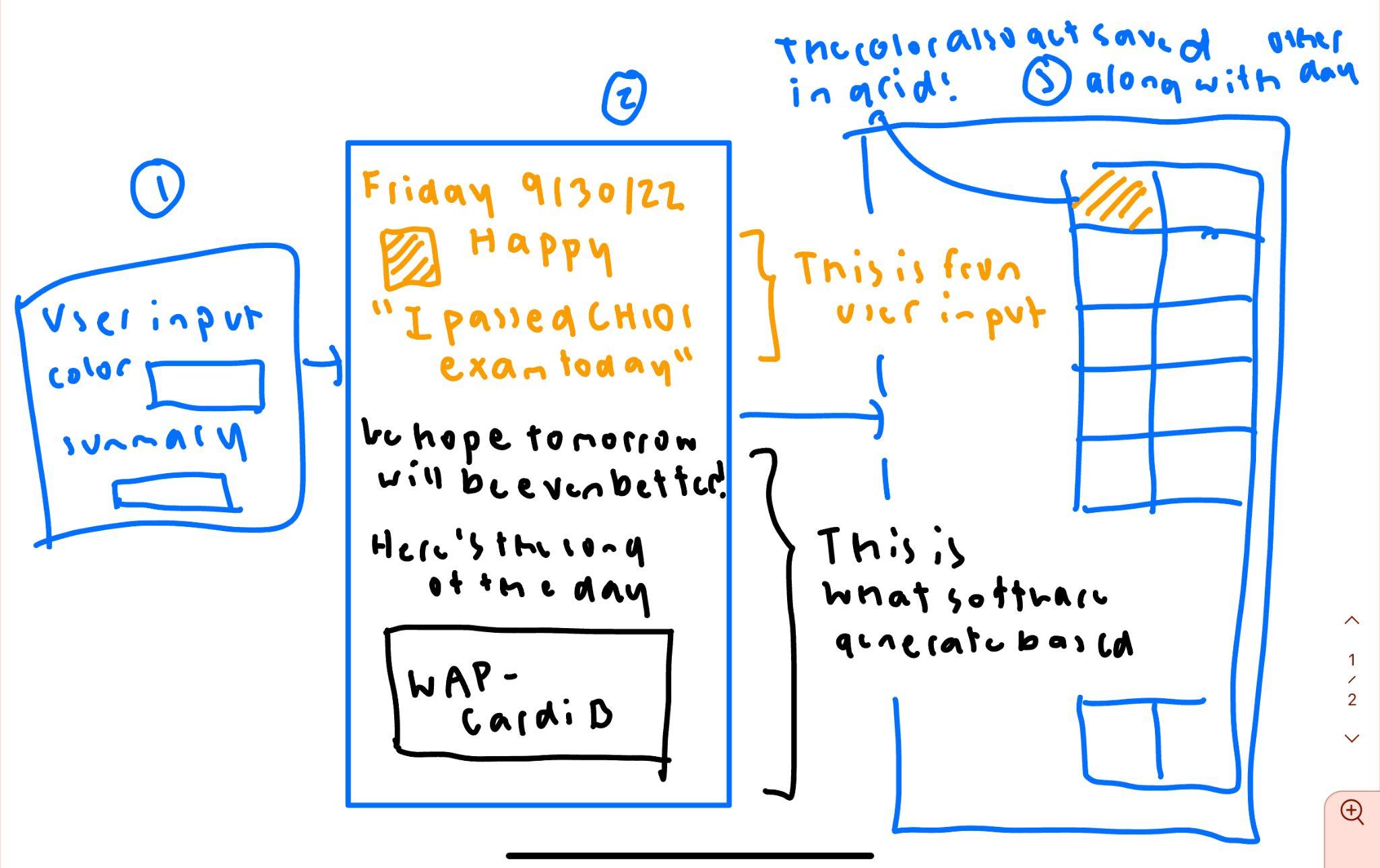
**Deliverables Document**

**Mood color**:

The user will create an account with their likes, hobbies, passions, etc. Once a user has an account their only data input will be a color that they believe best summarizes their day. We will pre-assign moods to be associated with each color. The color that the user feels associated with the day (like orange = energetic, yellow = happy etc.) will then be associated with the user’s pre-set likes and dislikes. Say a user enjoys smoothies and has a “yellow” day. We may suggest that the user gets a smoothie to stay in that mood. If the user had a blue day and likes rock music we could suggest them to a playlist from Spotify of slow or gloomy rock songs. The software would generate an entry (into shareable media, like GIF/PNG/JPEG) of the color the user chose and the summary of the day in one sentence. This summary would be a relationship between the day and the mood or may just tell the user a pre built statement. An example would be if it was a sunny day and the user chose their mood as orange, we may say “Hope the weather keeps it up!”. Finally, the colors the user chooses also becomes a grid where it stores choices of previous days; you can look at the big picture of what your week/year looks like.

Requirements:

* APIs: [Spotify Web API](https://developer.spotify.com/documentation/web-api/quick-start/), [Open Weather Map AP](https://openweathermap.org/api)I, more to come
* DB: MongoDB for user-by-user data tracking
* Front: Javascript - React for simplicity and capability
* Back: Node.js - Express
* Third Party Authentication: OAuth (Authentication API)



I might have to redraw this cause this is kinda old lol

**Price scanner:**

Scans for the prices across different platforms to find the cheapest price. User searches up for a specific item or category of item. API scans for the item of best fit across a bunch of different platforms. Users can sign in and choose which brands or companies they shop at most. They can also choose a price range that they are most comfortable with.

We shall use an API for comparing prices. Software searches from these companies, and other parameters, and chooses the most affordable option. The result of the search is a list of the specified item with comparisons across different companies.

Third party authentication - ebay, amazon

Requirements:

* APIs: [Rainforest API](https://get.rainforestapi.com/product/?gclid=Cj0KCQjwhY-aBhCUARIsALNIC05kE9C2JZshW6gOhGG_KASlYB5FcDxAhU9R9yafHUzQhBaNZAz1XF4aApl9EALw_wcB) (scrapes amazon product data), [RedCircle API](https://www.redcircleapi.com/docs/target-product-data-api/overview) (scrapes target); Ebay API
* DB: MongoDB for user-by-user data tracking
* Front: Javascript - React for simplicity and capability
* Back: Node.js - Express