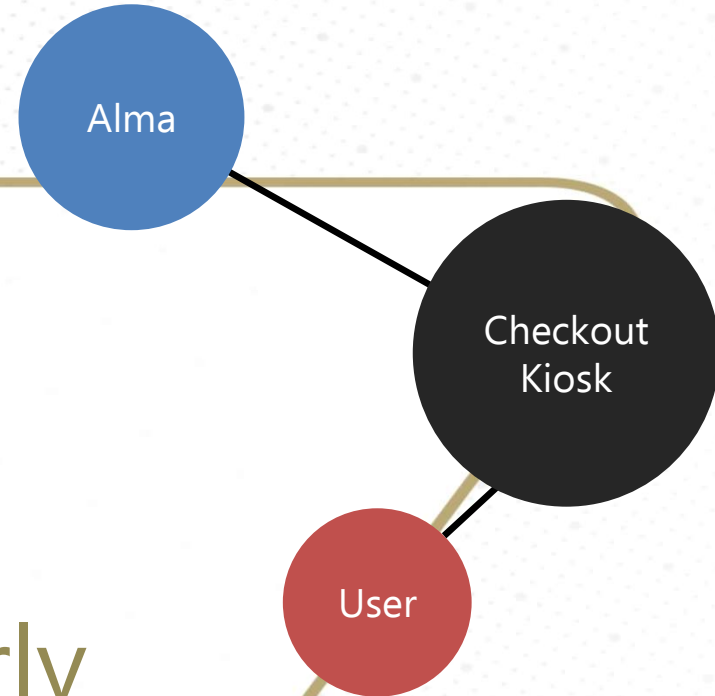


A Changing Culture: Innovative Technology to Improve the Scholarly Experience

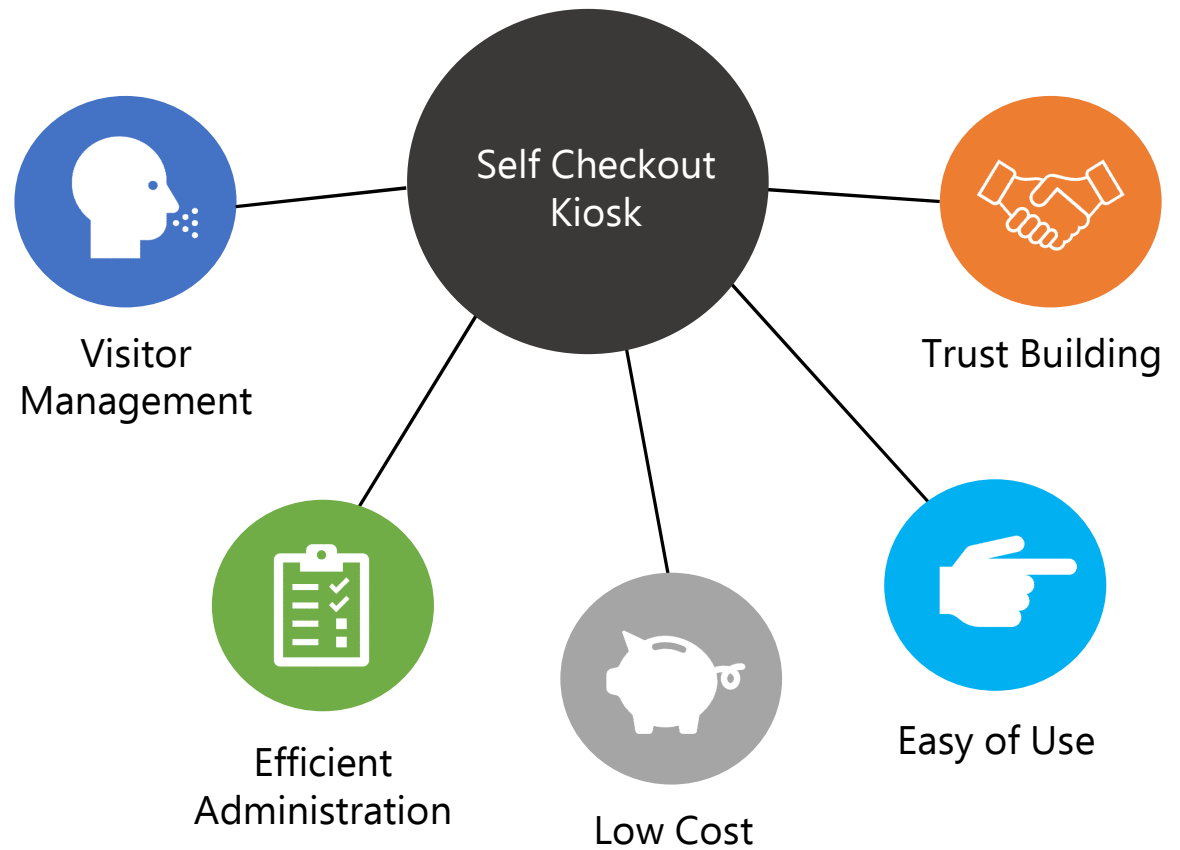
Low Price Self Checkout Kiosk

Intuitive to use | Easy to implement | Simple to manage

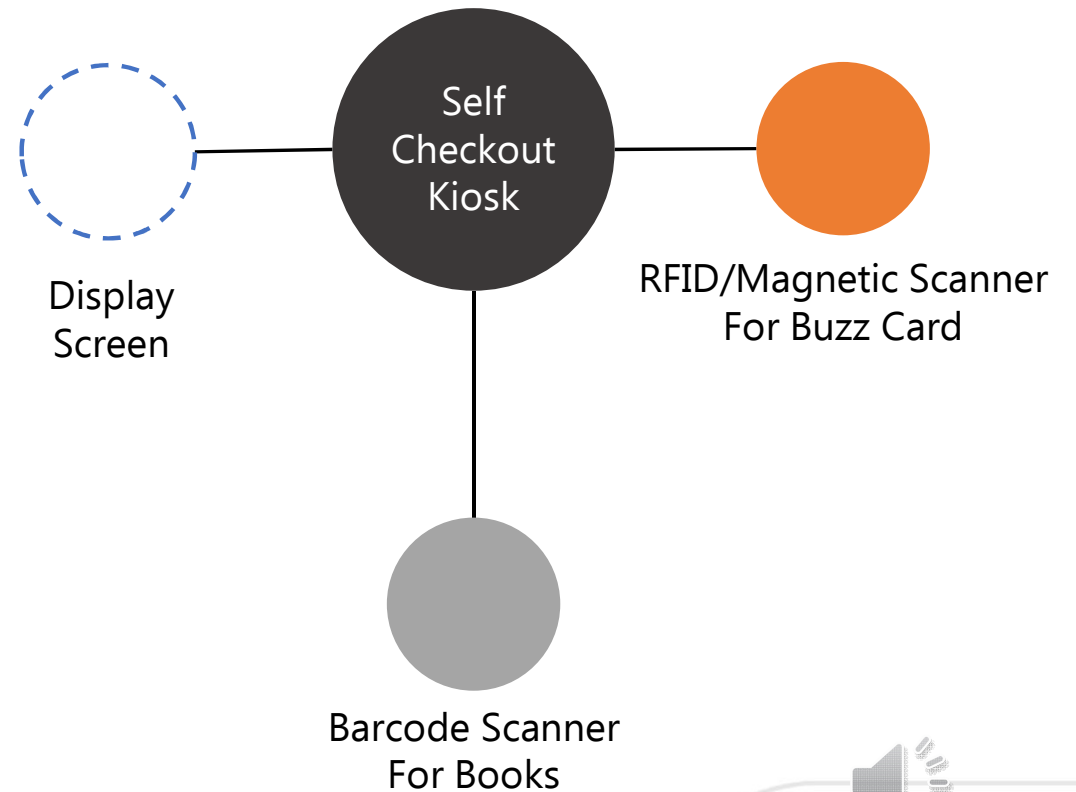


Good reasons

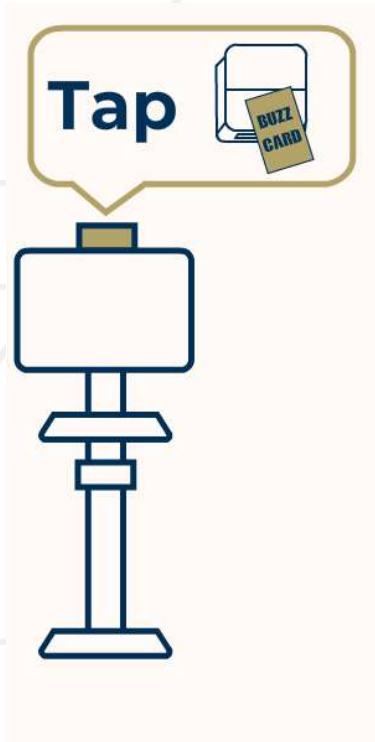
- Trust building
- Covid-19 secure visitor management
- Efficient and accurate administration
- Low cost
- Easy to implement, maintain, and use



Hardware – Main components



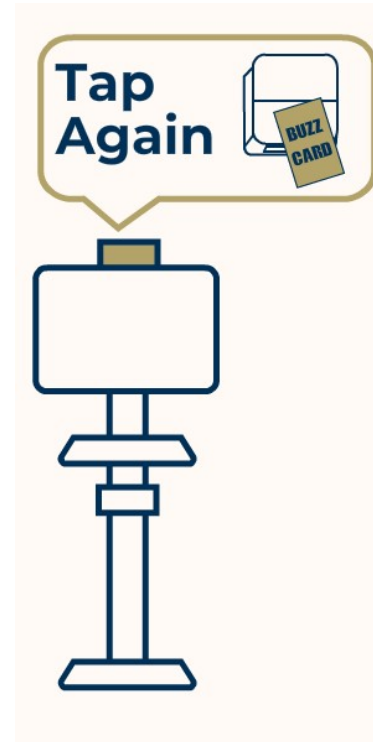
Overview – Workflow



Step 1
Tap Buzz card



Step 2
Scan books



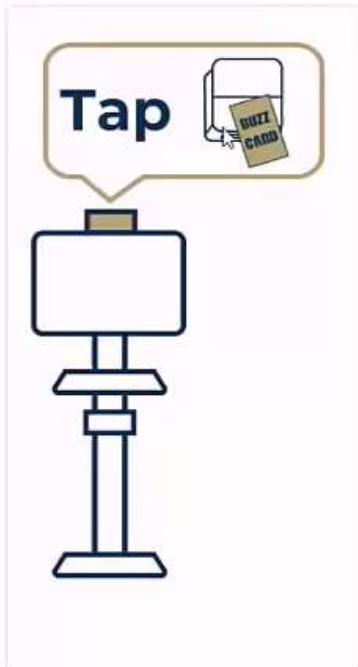
Step 3
Tap Buzz card
again
to complete

DEMO



Georgia Tech
Library

Self Checkout



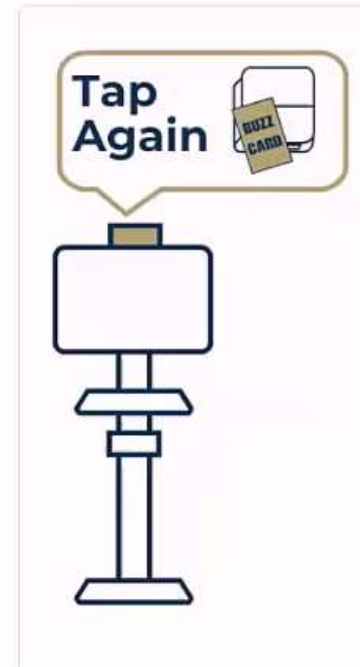
Step 1

Tap Buzz card to start



Step 2

Scan book barcode



Step 3

Tap Buzz card to complete

GUI: Form Plus Framework

Form Framework with Validation & Calculation

Forms Plus

Forms Plus is a form framework. **JS version** includes everything **CSS** has, plus date/time pickers, color pickers, sliders, captcha fields, spinners, field groups (for code, credit card number, etc.),

masked fields, actions blocks (add/remove/show/hide/toggle blocks with buttons, links, checkboxes, selects), image uploads, checkboxes and radio buttons (with toggle all, untoggle, groups and subgroups, etc.), and lots of features done with power of **Javascript**, **jQuery**, **CSS** and **PHP**.

Forms Plus has collection of contact, user, ecommerce forms, example of results page and allows you to create forms of any complexity.

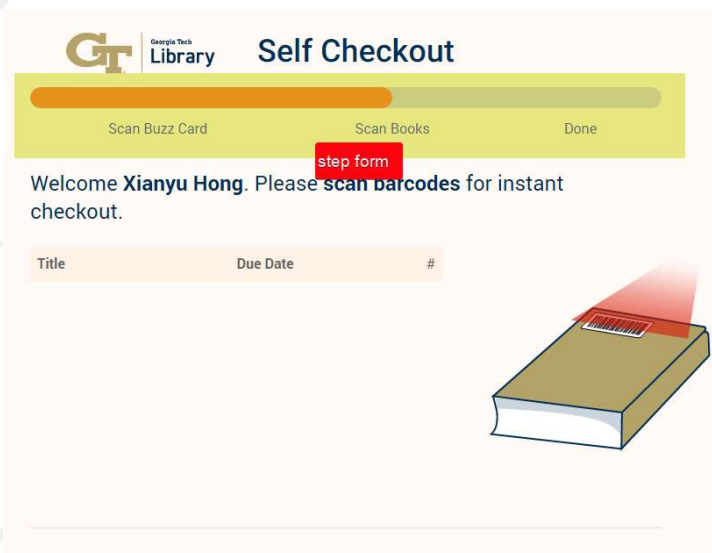
This product includes CSS, HTML, PHP, and Javascript.



GUI: Form Plus Framework

Self checkout APP uses

- Step form (1-2-3 steps) <https://apps-stage.library.gatech.edu/self-checkout/test/documentation/index.html#js-steps>
- Ajax - <https://apps-stage.library.gatech.edu/self-checkout/test/documentation/index.html#ajax>
- Blocks/Actions - <https://apps-stage.library.gatech.edu/self-checkout/test/documentation/index.html#blocks>



Where `<block name>` can be set on any tag with `data-js-block=""<block name>`

```
<!-- Start: modern-form - Ajax example -->
<form method="post" class="modern-p-form p-form modern-orange"
  action="/json/ajax-success.json" data-js-validate="true" data-js-highlight-state-msg="true" data-js-show-valid-msg="true"
  data-js-ajax-form=""
  data-js-ajax-before-show-block="loadingBlockName" data-js-ajax-before-hide-block="formInfoBlockName"
  data-js-ajax-success-show-block="successBlockName" data-js-ajax-success-hide-block="formInfoBlockName"
  data-js-ajax-fail-show-block="failBlockName" data-js-ajax-fail-hide-block="successBlockName"
  data-js-ajax-always-show-block="alwaysShowBlockName" data-js-ajax-always-hide-block="loadingBlockName">
</form class="p-form p-showed p-form">
```

Showing/hiding data blocks can be used to show/hide loading bars, hiding form on success, showing "thank you" messages, etc

Full JSON success response example:

```
{
  "block": "successContentBlock",
  "content": "Your success content!",
  "showBlocks": ["showBlockName1", "showBlockName2", ..., "showBlockNameN"],
  "hideBlocks": ["hideBlockName1", "hideBlockName2", ..., "hideBlockNameN"],
  "clearBlocks": ["clearBlockName1", "clearBlockName2", ..., "clearBlockNameN"],
  "fieldValues": [
    {
      "id": "field1-ld",
      "value": "field1 value"
    },
    {
      "id": "field2-ld",
      "value": "field2 value"
    },
    ...
    {
      "id": "fieldN-ld",
      "value": "fieldN value"
    }
  ]
}
```


GUI: Form Plus Framework

The screenshot shows the Georgia Tech Library Self Checkout interface. At the top left is the Georgia Tech Library logo. The main heading is "Self Checkout". Below the heading, there is a welcome message: "Welcome Xianyu Hong. Please scan barcodes for instant checkout." To the right of this message is an illustration of a book. A table below the message lists checked-out items. At the bottom of the page is a progress bar with the text "loading data, please wait...". Red boxes with arrows point to various elements: "error block" points to the logo and heading area; "info block" points to the welcome message; "user info. block" points to the user name "Xianyu Hong"; "main content block" points to the table; and "progress bar" points to the loading bar at the bottom.

GT Georgia Tech Library Self Checkout

error block
info block

error block

info block

user info. block

Welcome **Xianyu Hong**. Please **scan barcodes** for instant checkout.

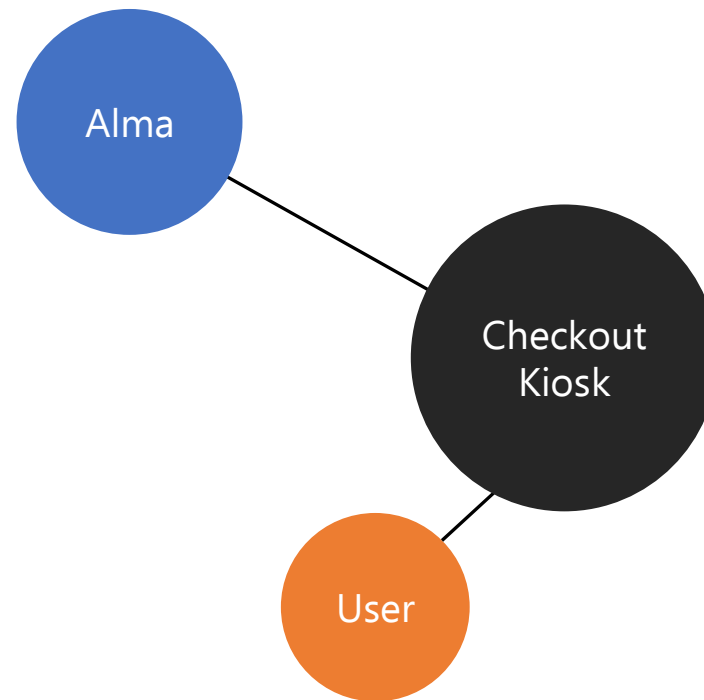
	Title	Due Date	#
✓ checked out	The best American comics. [50671014119140]	Sep 23, 2022	1

main content block

progress bar

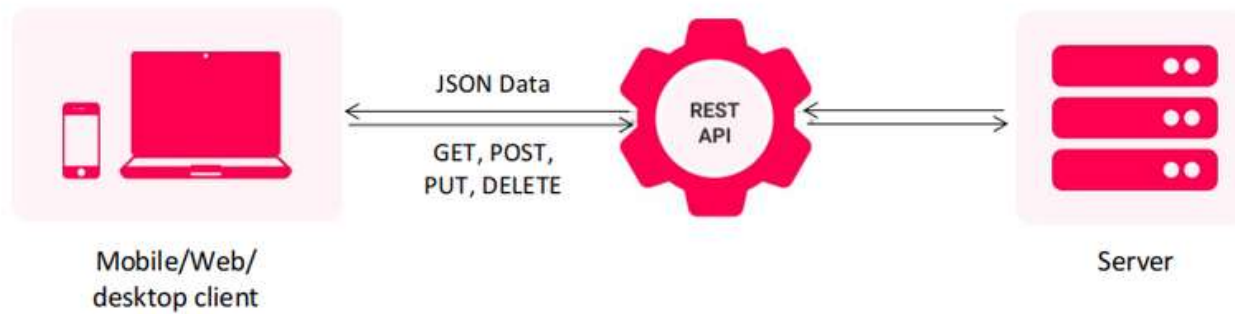
loading data, please wait...

Application - Overview

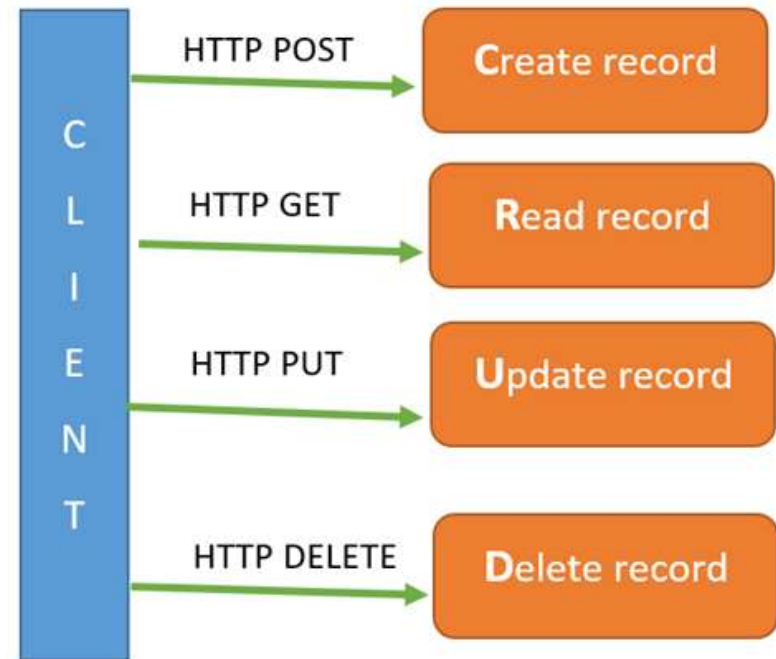
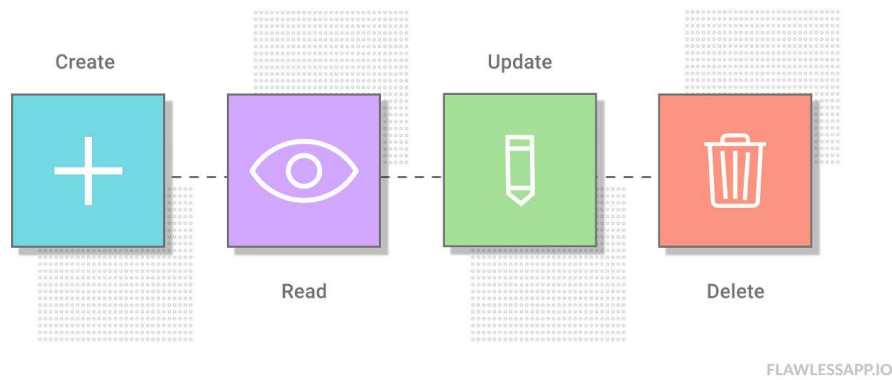


Application – REST & CRUD

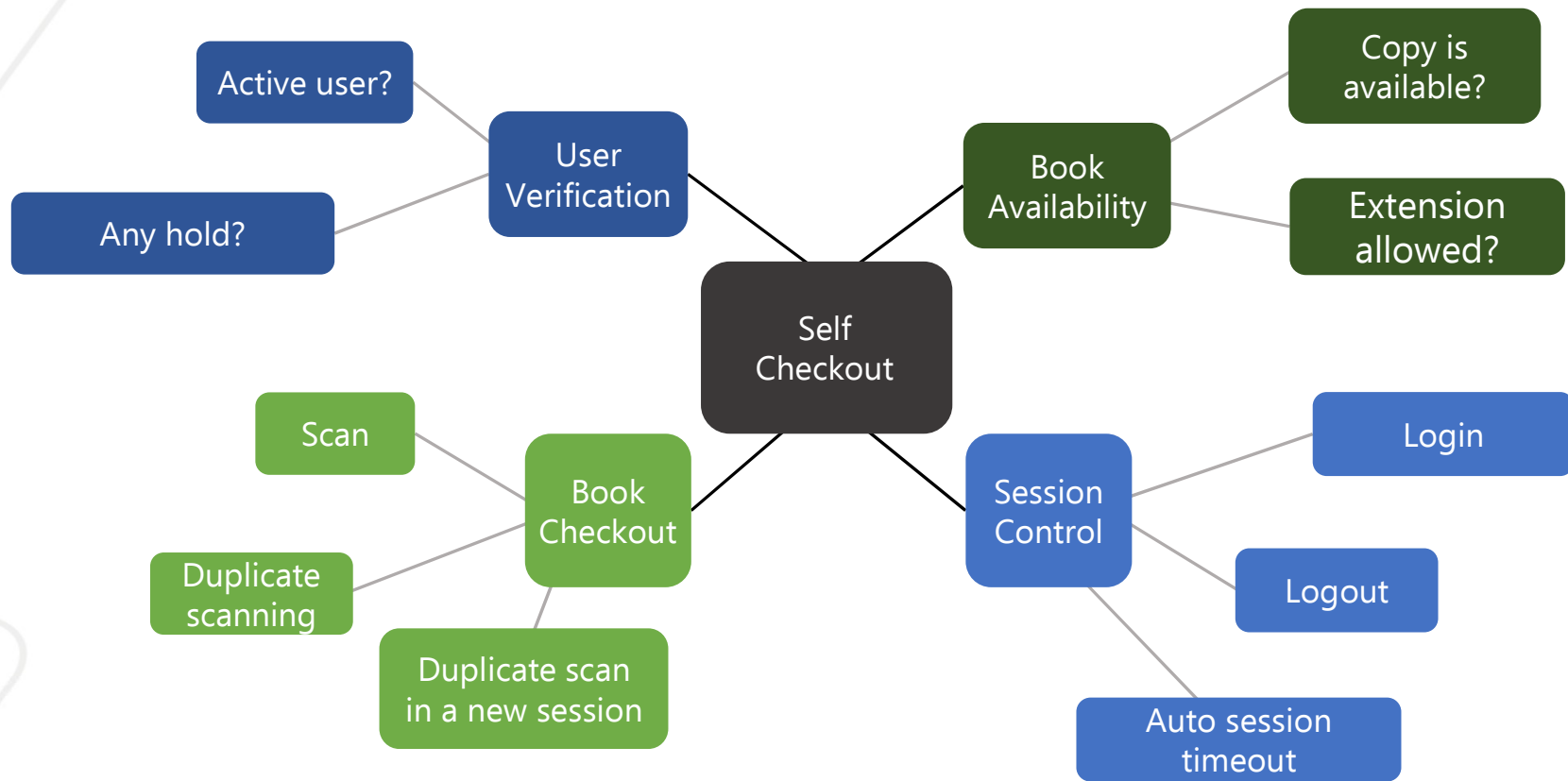
REST API Model



Application – REST & CRUD



Application - Main functions



Restful: HTTP Request Client

Requests for PHP

A humble HTTP request library. Simplify how you interact with other sites and leave your worries behind.

```
WpOrg\Requests\Requests::get('https://api.github.com/', array('Accept' => 'application/json'));
```

Never touch cURL again.

<https://requests.ryanmccue.info/>

GET Request

```
$response = WpOrg\Requests\Requests::get('https://api.github.com/events');
```

POST Request

```
$data = array('key1' => 'value1', 'key2' => 'value2');  
$response = WpOrg\Requests\Requests::post('https://httpbin.org/post', array(), $data);  
var_dump($response->body);
```

php curl print status

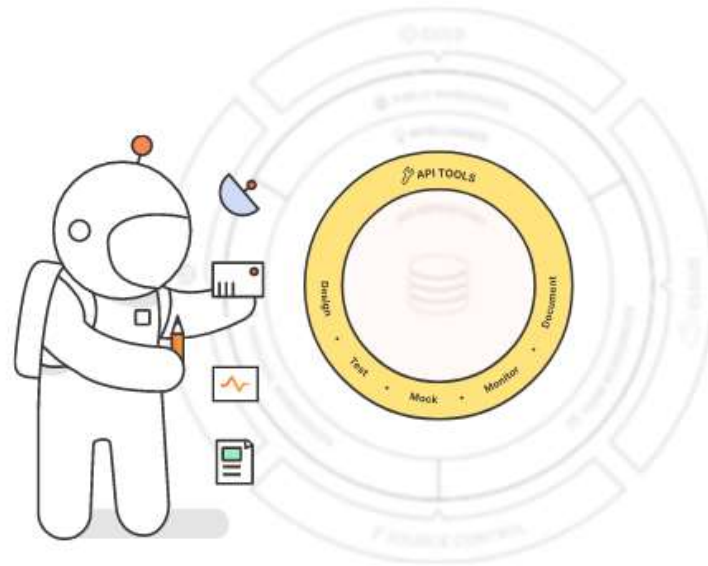
```
1 $url = 'https://api.github.com/events';  
2 $ch = curl_init($url);  
3 curl_setopt($ch, CURLOPT_HEADER, true); // we want headers  
4 curl_setopt($ch, CURLOPT_NOBODY, true); // we don't need body  
5 curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
6 curl_setopt($ch, CURLOPT_TIMEOUT, 10);  
7 $output = curl_exec($ch);  
8 $httpcode = curl_getinfo($ch, CURLINFO_HTTP_CODE);  
9 curl_close($ch);  
10  
11 echo 'HTTP code: ' . $httpcode;
```

Source: stackoverflow.com

get http code curl.php

```
1 $info = curl_getinfo($ch, CURLINFO_HTTP_CODE);  
2  
3 curl_close($ch);  
4 if ($info == 200){  
5     return json_decode($result);  
6 }
```

API Tool: Postman



<https://www.postman.com/>

API Tool: Postman Quick demo

Get users

Users

API

Path

Retrieve users

GET /almaws/v1/users

The image displays two screenshots of the Postman API client interface. The top screenshot shows the 'Retrieve users' endpoint with a 'Send' button highlighted. The bottom screenshot shows the 'Retrieve users' endpoint with the response body highlighted, displaying XML data for two users.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <users total_research_courses="107643"?>
3   <user link="https://api-na.hosted.exlibrisgroup.com/almaws/v1/users/98374388">
4     <primary_id>98374388</primary_id>
5     <first_name>Akansha</first_name>
6     <last_name></last_name>
7     <gender sec=""/><gender>
8     <password/><password>
9     <status sec="Active">ACTIVE</status>
10    <is_researcher>false</is_researcher>
11  </user>
12 <user link="https://api-na.hosted.exlibrisgroup.com/almaws/v1/users/98376310">
13   <primary_id>98376310</primary_id>
14   <first_name>Ajay</first_name>
```


Hardware design: Comparison



Questions?