



# **Graphical Password Authenticator**

## **Team UM7 Sprint 2**

**PM:** Tommy Lim, Nafis Hoque

**SM:** Nicholas Tatgenhorst, Rafaan Ahsan

**Programmers:** Ugonma Nnakwe, Eunice Olorunshola,

# Why use graphical passwords?

---

# Why use graphical passwords?



- Simplicity
- Better\* security

## Minimum Password Length

## Require at least one:

- ☒ Uppercase and lowercase letter (A, z)
- ☒ Numeric character (0-9)
- ☒ Special character (!, %, @, #, etc.)

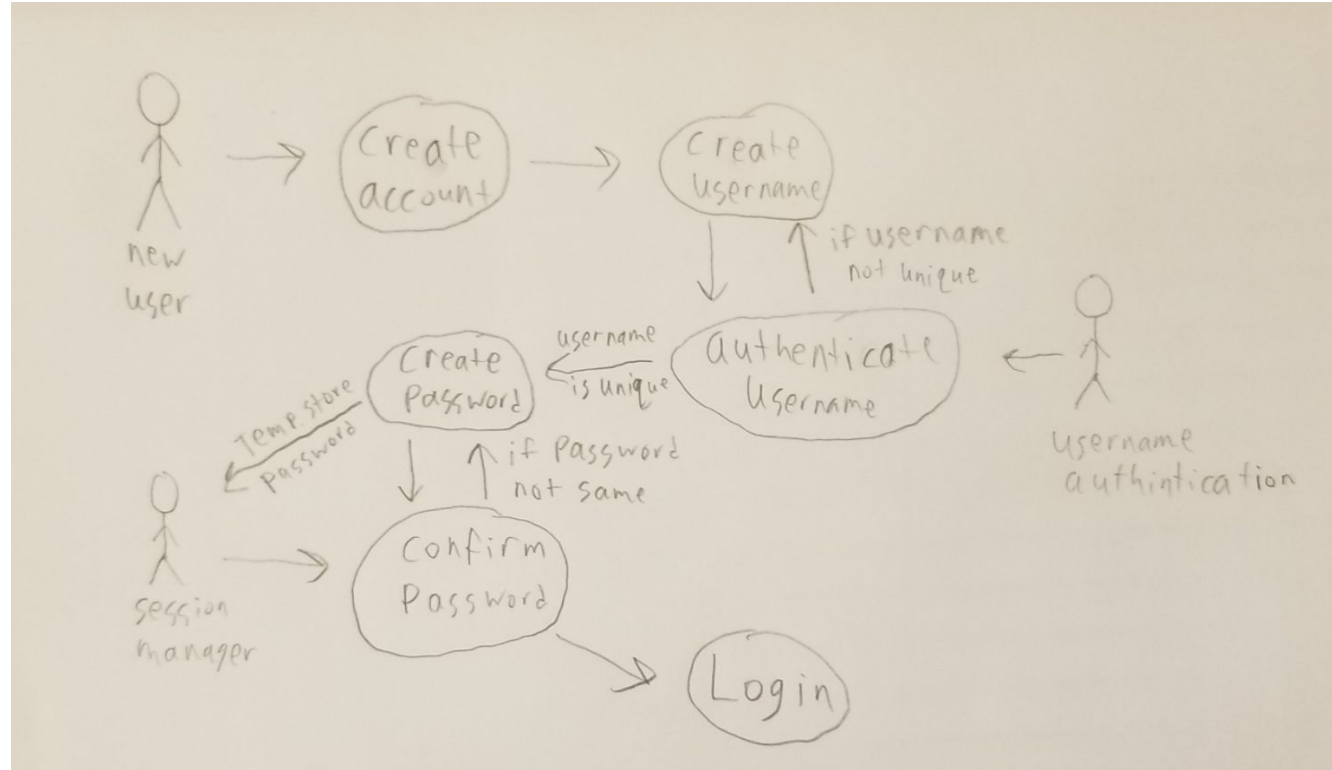


# UML Use-Case / Sequence Diagrams

---

## USE-CASE: New User

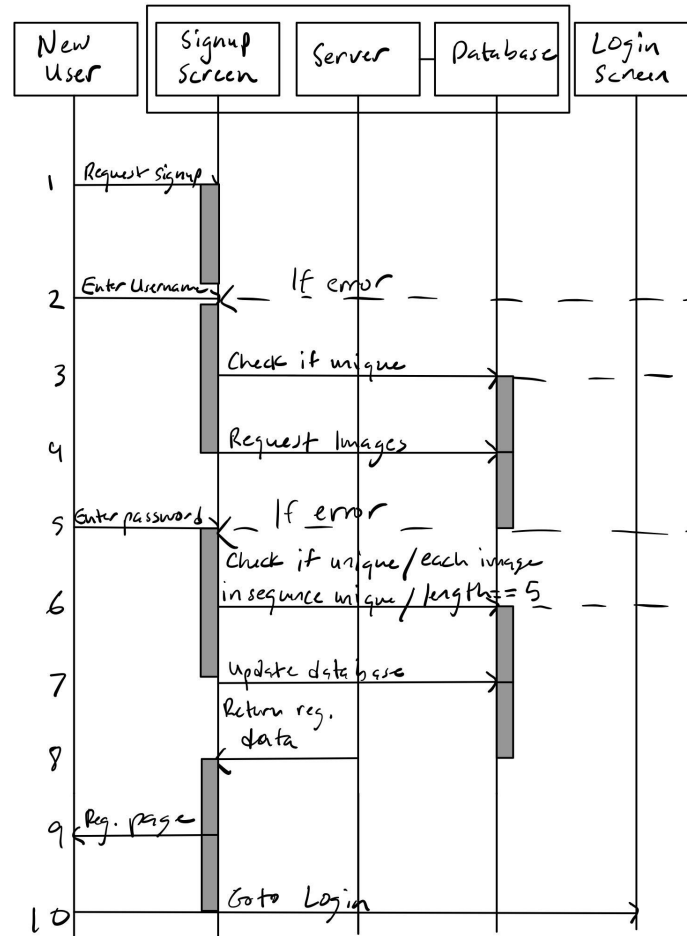
### New User



# SEQUENCE DIAGRAM: Registering a New User

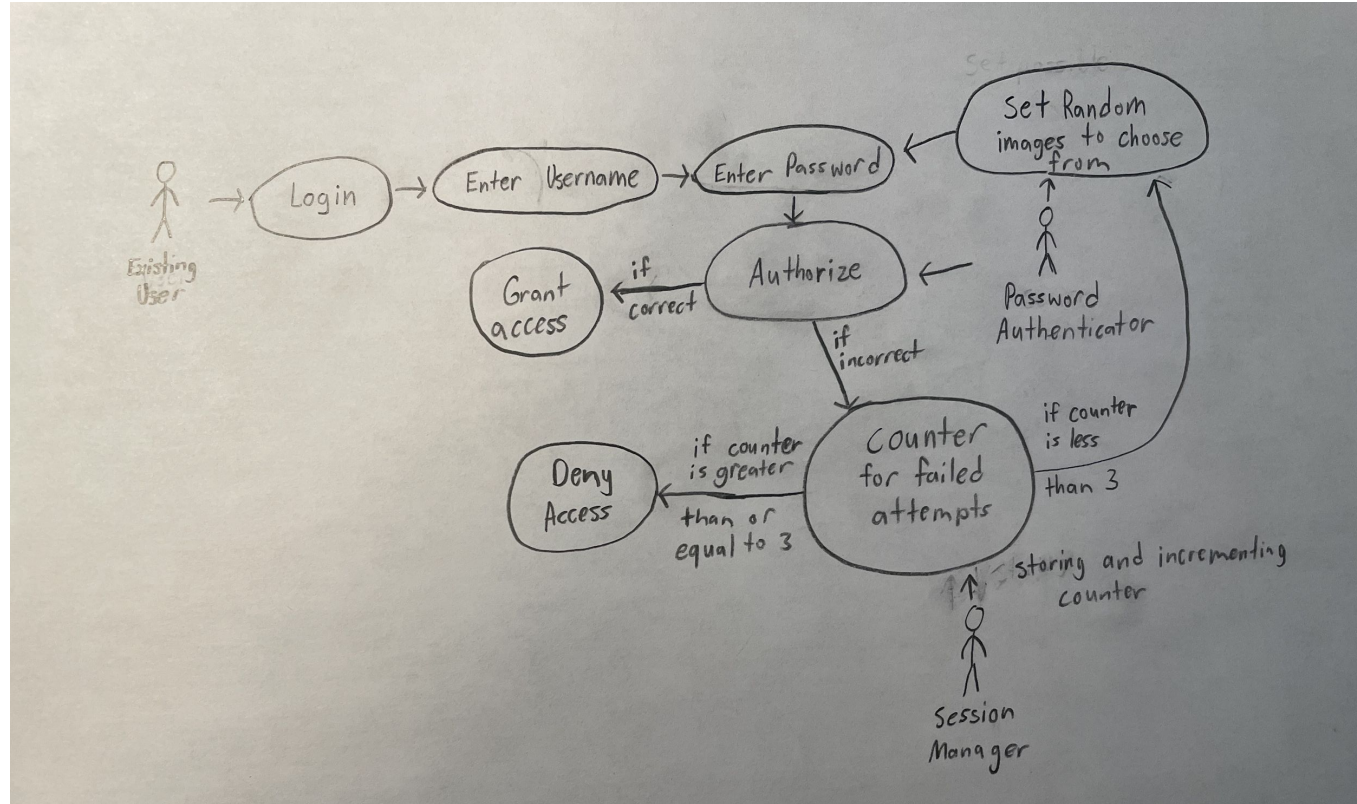
## Register New User

- Emphasize process of picking images\*

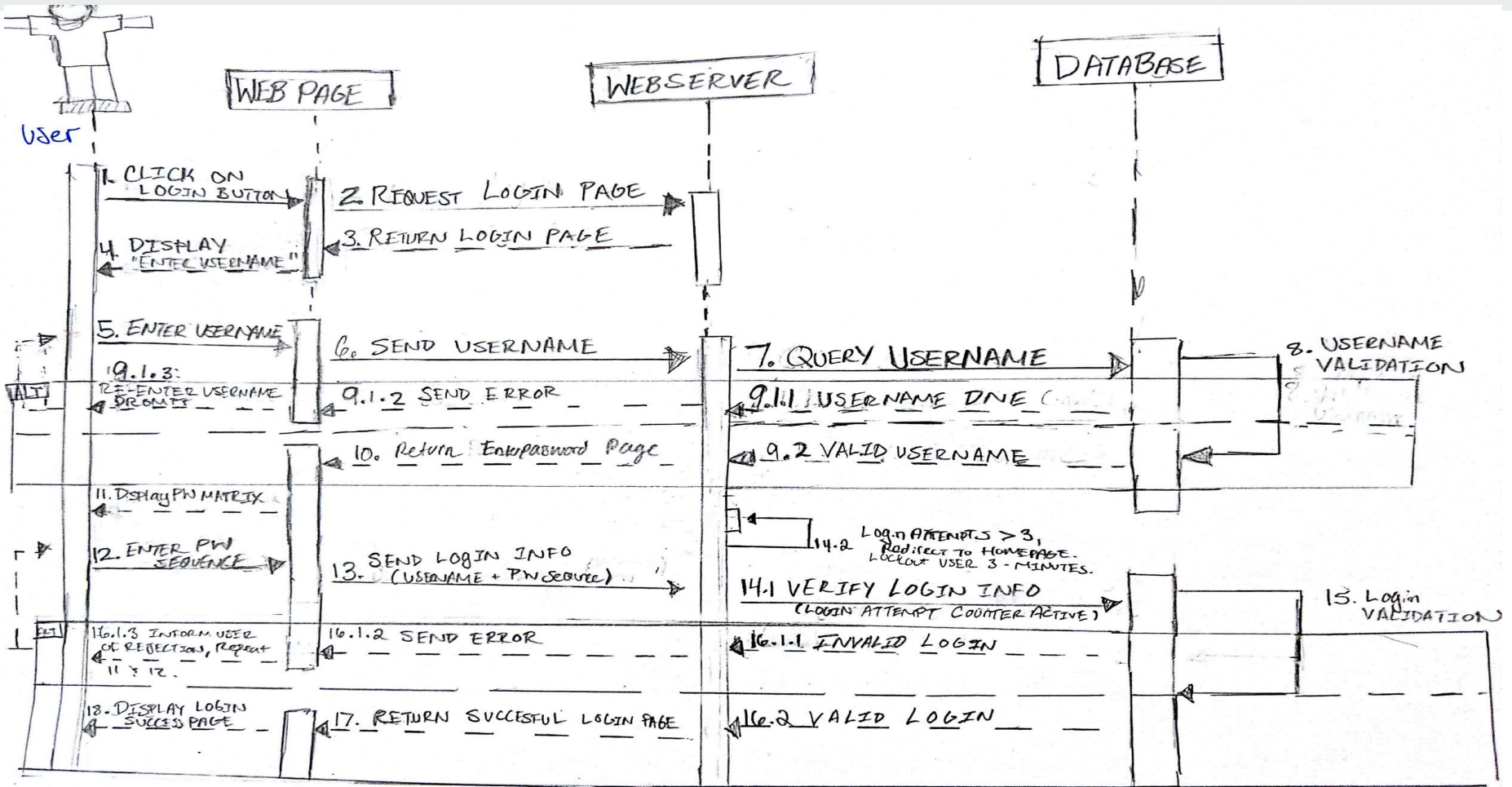


## USE-CASE: Existing User

### Existing User



# SEQUENCE DIAGRAM: Existing User Login Process





# Zenhub Components

---

# COMPLETED EPIC: UML Use-case Diagrams

UML Use Case Diagrams is an Epic

 Set start and end date

2 of 2 Issues completed

30 of 30 Epic Points completed

## Closed

 **Unsuccessful Login, and Lock-out of Registered User Use-Case**  
graphical-image-password-authenticator #35



Estimate

10

RafaanAhsan  
Ntatgenhorst



+1

Labels

 **Register New User Use-Case**  
graphical-image-password-authenticator #34


10



+1

# COMPLETED EPIC: UML Sequence Diagrams

UML Sequence Diagrams is an Epic

 Set start and end date

2 of 2 Issues completed

30 of 30 Epic Points completed

**Closed**

 **Register a New User Sequence**  
graphical-image-password-authenticator #32



Estimate

10

nhoque2  
thomadeus

 +1

Labels

 **Unsuccessful and Lockout Sequence**  
graphical-image-password-authenticator #33

10

 +1

# COMPLETED EPIC: Presentation Prep

Sprint 2 Presentation is an Epic

Mar 9 - Mar 22

3 of 3 Issues completed

41 of 41 Epic Points completed

## Closed

### Presentation Prep

graphical-image-password-authenticator #40

### Presentation Charts

graphical-image-password-authenticator #38

### Minimal Working Prototype of UX/UI

graphical-image-password-authenticator #39

## Estimate

5

5

21

## Assignees

 +5

ugo-n  
EuniceOlorunshola

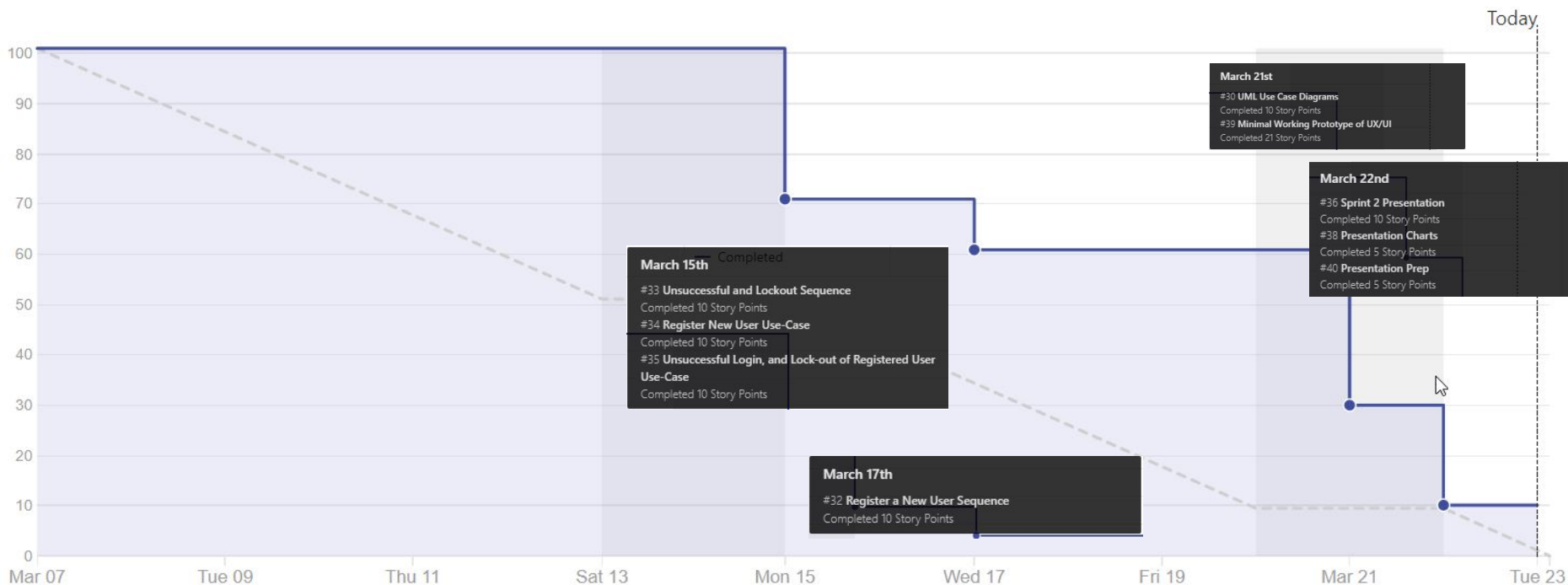
 +1

## Labels

# Burndown Chart

---



















# Burndown Chart



# Burndown Chart

## Completed Issues and Pull Requests

Story points

 <b>UML Use Case Diagrams</b> <span>Epic</span>	(10)
graphical-image-password-authenticator #30 - 000 Closed  Sprint 2	
 <b>Register a New User Sequence</b>	(10)
graphical-image-password-authenticator #32 - 000 Closed  Sprint 2	
 <b>Unsuccessful and Lockout Sequence</b>	(10)
graphical-image-password-authenticator #33 - 000 Closed  Sprint 2	
 <b>Register New User Use-Case</b>	(10)
graphical-image-password-authenticator #34 - 000 Closed  Sprint 2	
 <b>Unsuccessful Login, and Lock-out of Registered User Use-Case</b>	(10)
graphical-image-password-authenticator #35 - 000 Closed  Sprint 2	
 <b>Sprint 2 Presentation</b> <span>Epic</span>	(10)
graphical-image-password-authenticator #36 - 000 Closed  Sprint 2	
 <b>Presentation Charts</b>	(5)
graphical-image-password-authenticator #38 - 000 Closed  Sprint 2	
 <b>Minimal Working Prototype of UX/UI</b>	(21)
graphical-image-password-authenticator #39 - 000 Closed  Sprint 2	
 <b>Presentation Prep</b>	(5)
graphical-image-password-authenticator #40 - 000 Closed  Sprint 2	

## Incomplete Issues and Pull Requests that were not completed by the end of the Milestone

Story points

 <b>Determine DB Attributes</b>	(5)
graphical-image-password-authenticator #5 - 000 Sprint Backlog  Sprint 2	
 <b>Draft SQL tables for user and password information</b>	(5)
graphical-image-password-authenticator #19 - 000 Sprint Backlog  Sprint 2	

# Burndown Chart



Incomplete Issues and Pull Requests that were not completed by the end of the Milestone

Story points



**Determine DB Attributes**

graphical-image-password-authenticator #5 - 000 Sprint Backlog ↗ Sprint 2

5



**Draft SQL tables for user and password information**

graphical-image-password-authenticator #19 - 000 Sprint Backlog ↗ Sprint 2

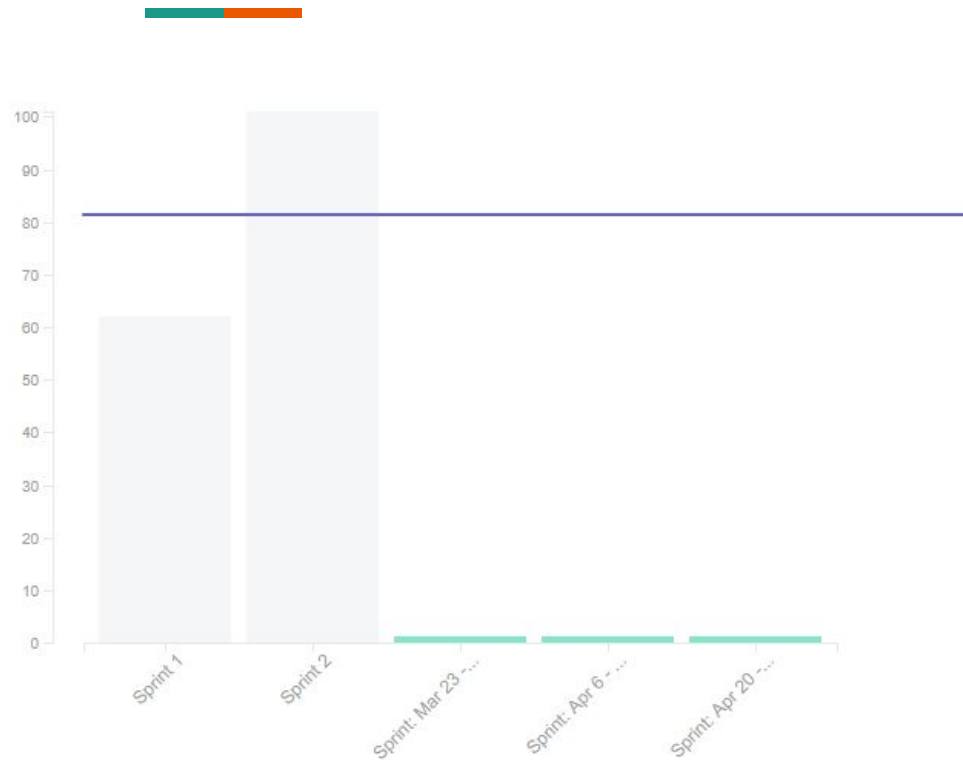
5



# Velocity Chart

---

# Velocity Chart



## Sprint 2

91 of 101 story points completed

Mar 7, 2021

Start Date

Mar 23, 2021

End Date

## Average Velocity

76.5 average velocity over closed sprints

2

Closed

153

Story Points

## Sprint 1

62 of 72 story points completed

Feb 6, 2021

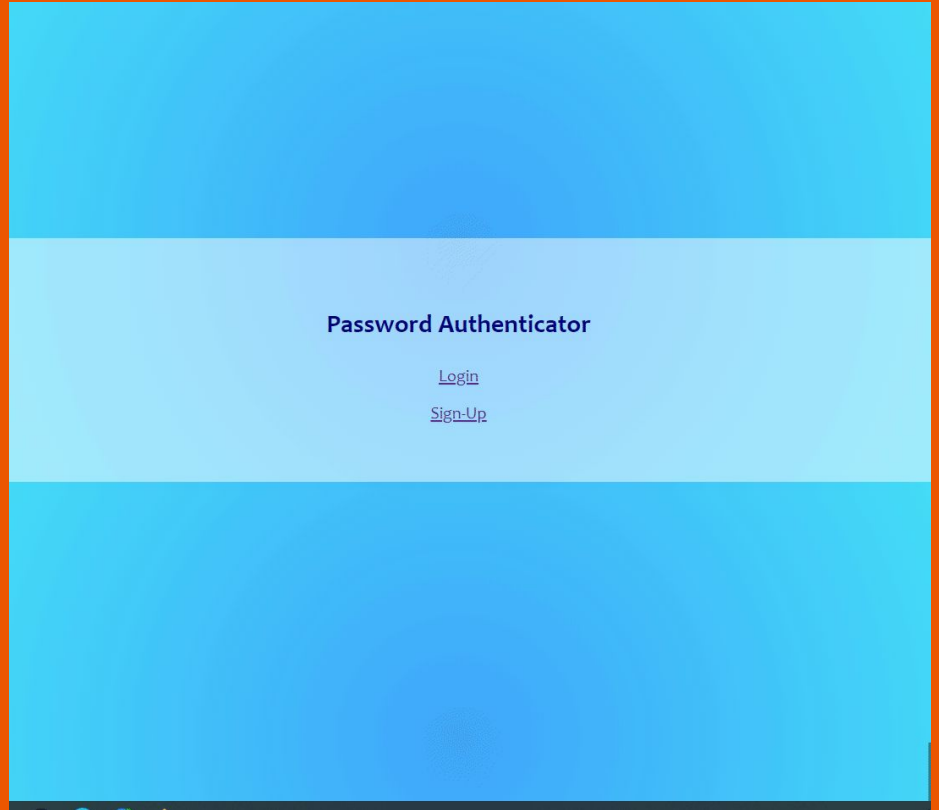
Start Date

Feb 25, 2021

End Date

# UI Prototype

---



# Screenshots from create-password.php

Retrieving  
data from  
php to  
javascript  
(image  
names)

```
<?php
include 'common.php';
$foodnames = file_get_contents("files.txt");
$animalnames = file_get_contents("animals.txt");
$transportnames = file_get_contents("transport.txt");
$foodarray = explode("\n", $foodnames);
$animalarray = explode("\n", $animalnames);
$transportarray = explode("\n", $transportnames);
?>

<!DOCTYPE html>
<html Lang = "en">
<head>
    <title>Password Prototype</title>
    <link rel="stylesheet" href="prototype.css">

    <!--script for generating images-->
    <!--needs to be cleaned,-->
    <script type="text/javascript">
        var type = 1;
        var animalArray =
            <?php echo json_encode($animalarray); ?>;
        var tranportArray =
            <?php echo json_encode($transportarray); ?>;

        var foodArray =
            <?php echo json_encode($foodarray); ?>;
```

Creating  
image-grids  
(lists) by  
retrieving  
images from  
their paths

```
function makeList(num){
    //make List based on start
    var listData,
    listContainer = document.getElementById('image-select-grid'),
    listItem = document.getElementById('image-list'),
    listItem,
    i,
    startPath = "icons/",
    type = num;
    //make conditional based on startPath
    switch(type){
        case 1:
            listItem = document.getElementById('image-list'),
            listData = foodArray;
            break;
        case 2:
            listItem = document.getElementById('t-image-list'),
            listData = tranportArray;
            break;
        case 3:
            listItem = document.getElementById('a-image-list'),
            listData = animalArray;
            break;
        default:
            break;
    }

    var numberOfListItems = listData.length;
    for(i = 0; i < numberOfListItems; ++i){
        listItem = document.createElement('li');

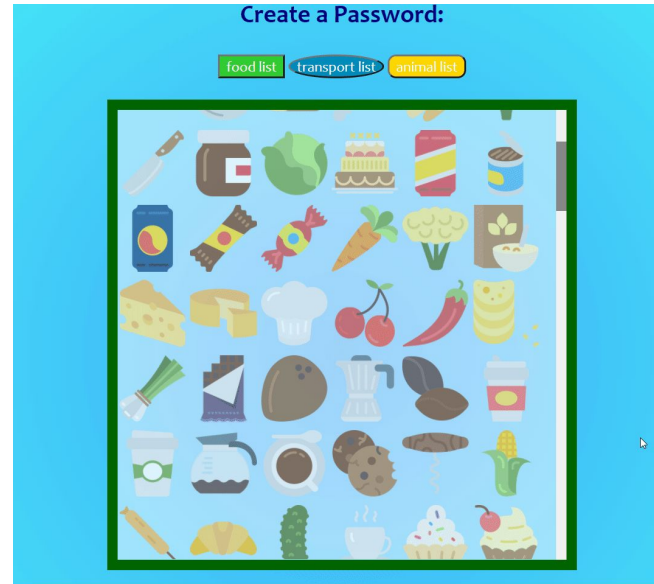
        listItem.innerHTML = "<img src='" + startPath + listData[i] + "'>" + "</img>";

        listItem.appendChild(listItem);
    }
}
```

# Screenshots

Function to choose which images are displayed for password creation

```
function showAnimal(){
    document.getElementById("a-image-list").style.display = "block";
    document.getElementById("t-image-list").style.display = "none";
    document.getElementById("image-list").style.display = "none";
}
function showFood(){
    document.getElementById("a-image-list").style.display = "none";
    document.getElementById("t-image-list").style.display = "none";
    document.getElementById("image-list").style.display = "block";
}
function showTransport(){
    document.getElementById("a-image-list").style.display = "none";
    document.getElementById("t-image-list").style.display = "block";
    document.getElementById("image-list").style.display = "none";
}
```

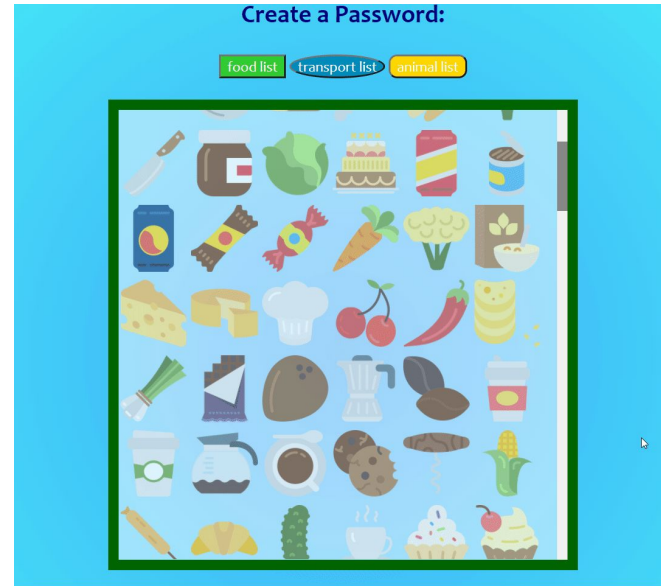


# Screenshots

```
<div id="image-select-grid">
  <ul id="image-list">
    <!--generate list in js of all icons for category-->
  </ul>
  <ul id="a-image-list">
  </ul>
  <ul id="t-image-list">
  </ul>
</div>

<h2>Chosen Images:</h2>
<div id="password-display">
  <ul id="images-selected">
    <li id = "image-1"></li>
    <li id = "image-2"></li>
    <li id = "image-3"></li>
    <li id = "image-4"></li>
    <li id = "image-5"></li>
  </ul>
</div>
```

Using unordered lists  
to display the images



## Screenshots from enter-password.php

```
<?php
include 'common.php';

$foodnames = file_get_contents("files.txt");
$animalnames = file_get_contents("animals.txt");
$transportnames = file_get_contents("transport.txt");
$foodarray = explode("\n", $foodnames);
$animalarray = explode("\n", $animalnames);
$transportarray = explode("\n", $transportnames);

$all_images = array_merge($foodarray, $animalarray, $transportarray)

?>

<!DOCTYPE html>
<html Lang = "en">
<head>
    <title>Password Prototype</title>
    <link rel="stylesheet" href="prototype.css">

    <!--script for generating images-->
    <!--needs to be cleaned,-->
    <script type="text/javascript">
        var images_array =
            <?php echo json_encode($all_images); ?>;
```

## Retrieving data from php to javascript

# Screenshots from enter-password.php

```
function generateList()  
{  
    var image_bank = getRandom(images_array, 36), listItem = document.getElementById('enter-list'),  
        listItem,  
        startPath = "icons/",  
        i;  
  
    var numberOfListItems = image_bank.length;  
  
    for(i = 0; i < numberOfListItems; ++i){  
        listItem = document.createElement('li');  
  
        listItem.innerHTML = "<img src='" + startPath + image_bank[i] + "'>" + "</img>";  
  
        listItem.appendChild(listItem);  
    }  
}  
  
//code referenced from Bergi  
//link: https://stackoverflow.com/questions/19269545/how-to-get-a-number-of-random-elements-from-an-array  
function getRandom(arr, n) {  
    var result = new Array(n),  
        len = arr.length,  
        taken = new Array(len);  
    if (n > len)  
        throw new RangeError("getRandom: more elements taken than available");  
    while (n--) {  
        var x = Math.floor(Math.random() * len);  
        result[n] = arr[x in taken ? taken[x] : x];  
        taken[x] = --len in taken ? taken[len] : len;  
    }  
    return result;  
}  
//end code referenced from bergi  
//link: https://stackoverflow.com/questions/19269545/how-to-get-a-number-of-random-elements-from-an-array
```

Creating randomly  
generated image grid  
for entering password



## Screenshots from enter-password.php


```
<body onLoad="generateList()">
  <h2>Enter Password:</h2>
  <div id = "password-enter">
    <ul id="enter-list">
      <!--generate list of random icons-->
    </ul>
  </div>
  <form action="login-confirm.php"
        method="post">
    <input type="submit" value="Login"/>
  </form>
</body>
```

Generates random list of images on load of the page.

# Screenshots

Create a Username:

Create a Password:



Chosen Images:

Enter Password:





## Link to prototype on CODD Server

Link: <https://codd.cs.gsu.edu/~unnakwe1/prototype/index.php>

No selection of images yet, still in the works.

Images take some time to load and load visibly. May add some sort of visual loading images before showing image grids to give the web-page time to load them all.

Not exactly the exact implementation we want. Moreso wanted variations of the same group of images rather than a bunch of unique images.



## Plans for Sprint 3

Solidify Database usage-still having difficulty implementing it due to lack of experience.

Create functional password hashing algorithm for additional security in the implementation

Create the functions necessary for username and password authentication.