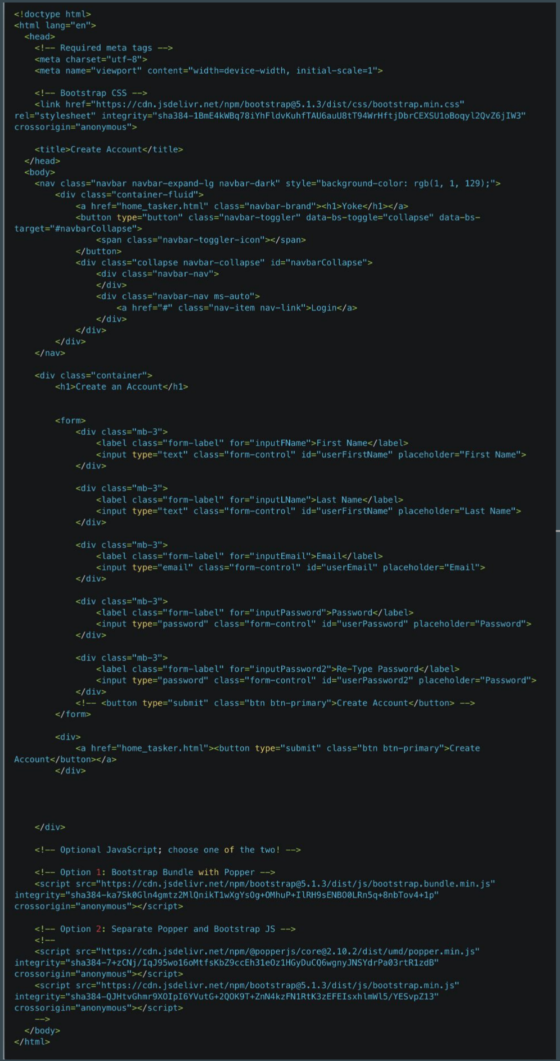
Yoke

Labor-Employment Administration App designed by (Group 8):
Michael Childs (Back-End, Documentation)
Brian Mathews (Front-End, QoL)
Jasper Swensen (Full-Stack, Testing)

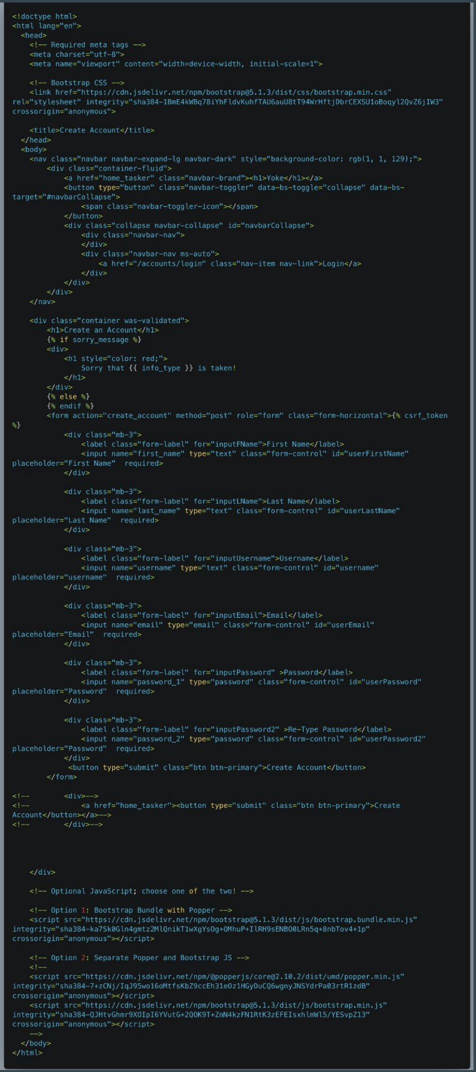
Yoke Overview

Yoke is an app for property-managers (known hereafter as “Taskers”) to employ freelance laborers (known hereafter as “Workers”) for various tasks and projects on local properties. Yoke is both desktop and mobile compatible and employs a payment system to accompany the seamless task-creation and queuing system. Team 8 worked together to make various design decisions with respect to the app’s development. Each member’s strengths were utilized to the fullest- Brian focused on front-end development, such as CSS optimization and general Bootstrap page-template design. Jasper concentrated on full-stack production, designing a refined database schema for Django while also supporting front-end functionality and services. Michael took charge of maintaining the project’s documentation, appointment as primary Scrum-Master, and back-end implementation assistance to Jasper.

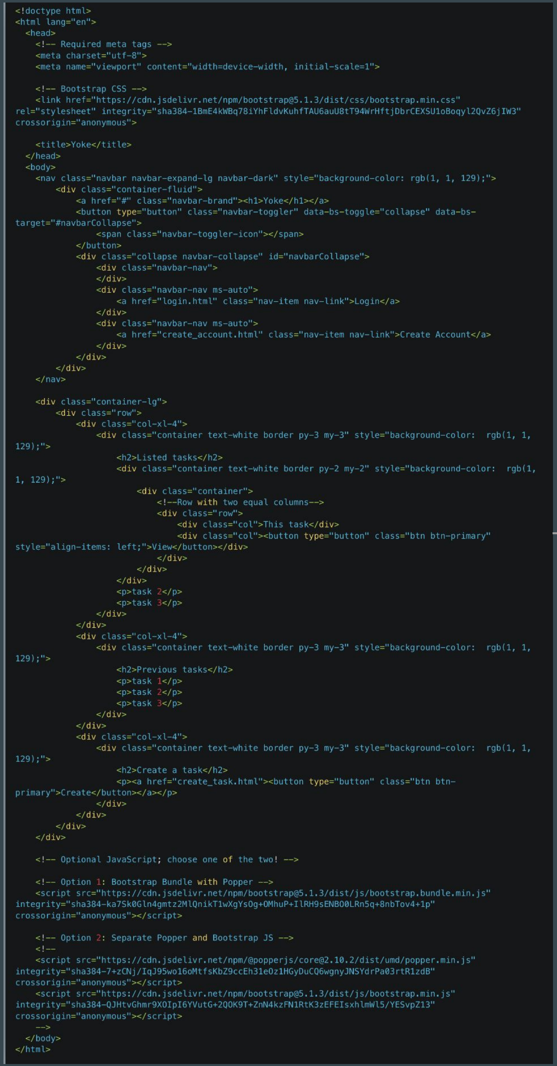
Code Progression



```html
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

    <title>Create Account</title>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark" style="background-color: rgb(1, 1, 129);">
        <div class="container-fluid">
            <a href="home_tasker.html" class="navbar-brand"><h1>Yoke</h1></a>
            <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-
target="#navbarCollapse">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarCollapse">
                <div class="navbar-nav">
                </div>
                <div class="navbar-nav ms-auto">
                    <a href="#" class="nav-item nav-link">Login</a>
                </div>
            </div>
        </div>
    </nav>

    <div class="container">
        <h1>Create an Account</h1>


        <form>
            <div class="mb-3">
                <label class="form-label" for="inputFName">First Name</label>
                <input type="text" class="form-control" id="userFirstName" placeholder="First Name">
            </div>

            <div class="mb-3">
                <label class="form-label" for="inputLName">Last Name</label>
                <input type="text" class="form-control" id="userFirstName" placeholder="Last Name">
            </div>

            <div class="mb-3">
                <label class="form-label" for="inputEmail">Email</label>
                <input type="email" class="form-control" id="userEmail" placeholder="Email">
            </div>

            <div class="mb-3">
                <label class="form-label" for="inputPassword">Password</label>
                <input type="password" class="form-control" id="userPassword" placeholder="Password">
            </div>

            <div class="mb-3">
                <label class="form-label" for="inputPassword2">Re-Type Password</label>
                <input type="password" class="form-control" id="userPassword2" placeholder="Password">
            </div>
            <!-- <button type="submit" class="btn btn-primary">Create Account</button> -->
        </form>

        <div>
            <a href="home_tasker.html"><button type="submit" class="btn btn-primary">Create
Account</button></a>
        </div>




    </div>

    <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

    <!-- Option 2: Separate Popper and Bootstrap JS -->
    <!--
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"
integrity="sha384-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"
crossorigin="anonymous"></script>
    -->
  </body>
</html>
```

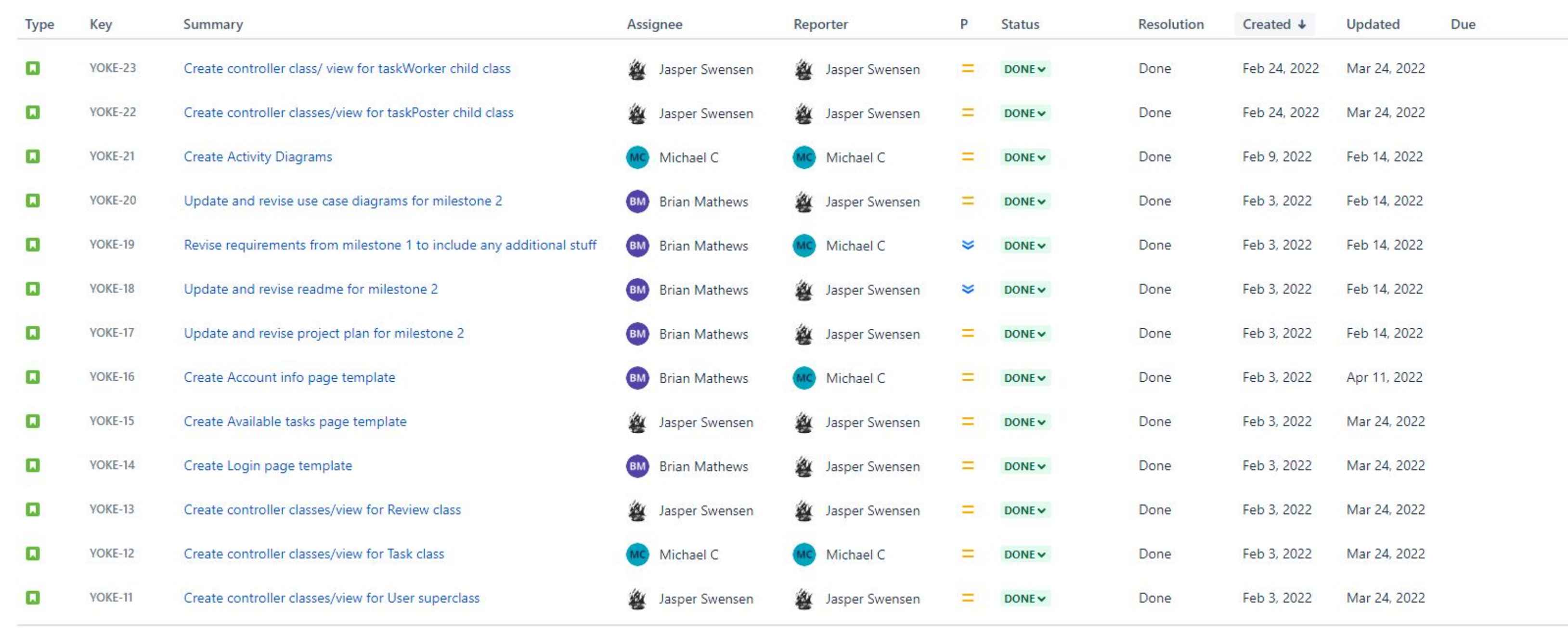


```html
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

    <title>Create Account</title>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark" style="background-color: rgb(1, 1, 129);">
        <div class="container-fluid">
            <a href="home_tasker" class="navbar-brand"><h1>Yoke</h1></a>
            <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-
target="#navbarCollapse">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarCollapse">
                <div class="navbar-nav">
                </div>
                <div class="navbar-nav ms-auto">
                    <a href="/accounts/login" class="nav-item nav-link">Login</a>
                </div>
            </div>
        </div>
    </nav>

    <div class="container was-validated">
        <h1>Create an Account</h1>
        {% if sorry_message %}
        <div>
            <h1 style="color: red;">
                Sorry that {{ info_type }} is taken!
            </h1>
        </div>
        {% else %}
        {% endif %}
        <form action="create_account" method="post" role="form" class="form-horizontal">{% csrf_token
%}
            <div class="mb-3">
                <label class="form-label" for="inputFName">First Name</label>
                <input name="first_name" type="text" class="form-control" id="userFirstName"
placeholder="First Name"  required>
            </div>

            <div class="mb-3">
                <label class="form-label" for="inputLName">Last Name</label>
                <input name="last_name" type="text" class="form-control" id="userLastName"
placeholder="Last Name"  required>
            </div>

            <div class="mb-3">
                <label class="form-label" for="inputUsername">Username</label>
                <input name="username" type="text" class="form-control" id="username"
placeholder="username"  required>
            </div>

            <div class="mb-3">
                <label class="form-label" for="inputEmail">Email</label>
                <input name="email" type="email" class="form-control" id="userEmail"
placeholder="Email"  required>
            </div>

            <div class="mb-3">
                <label class="form-label" for="inputPassword" >Password</label>
                <input name="password_1" type="password" class="form-control" id="userPassword"
placeholder="Password"  required>
            </div>

            <div class="mb-3">
                <label class="form-label" for="inputPassword2" >Re-Type Password</label>
                <input name="password_2" type="password" class="form-control" id="userPassword2"
placeholder="Password"  required>
            </div>
             <button type="submit" class="btn btn-primary">Create Account</button>
        </form>

<!--        <div>-->
<!--            <a href="home_tasker"><button type="submit" class="btn btn-primary">Create
Account</button></a>-->
<!--        </div>-->




    </div>

    <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

    <!-- Option 2: Separate Popper and Bootstrap JS -->
    <!--
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"
integrity="sha384-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"
crossorigin="anonymous"></script>
    -->
  </body>
</html>
```



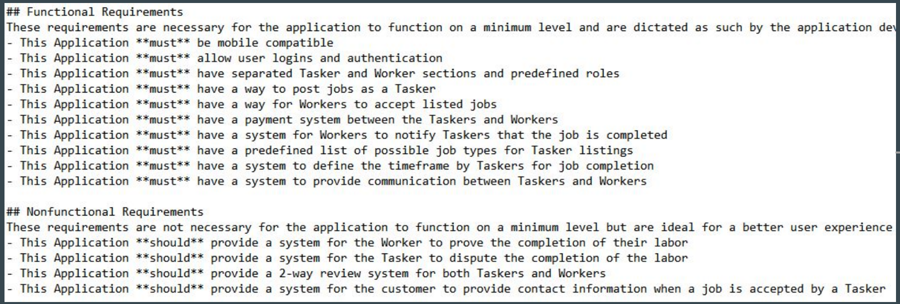
```html
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

    <title>Yoke</title>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark" style="background-color: rgb(1, 1, 129);">
        <div class="container-fluid">
            <a href="#" class="navbar-brand"><h1>Yoke</h1></a>
            <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-
target="#navbarCollapse">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarCollapse">
                <div class="navbar-nav">
                </div>
                <div class="navbar-nav ms-auto">
                    <a href="login.html" class="nav-item nav-link">Login</a>
                </div>
                <div class="navbar-nav ms-auto">
                    <a href="create_account.html" class="nav-item nav-link">Create Account</a>
                </div>
            </div>
        </div>
    </nav>

    <div class="container-lg">
        <div class="row">
            <div class="col-xl-4">
                <div class="container text-white border py-3 my-3" style="background-color:  rgb(1, 1,
129);">
                    <h2>Listed tasks</h2>
                    <div class="container text-white border py-2 my-2" style="background-color:  rgb(1,
1, 129);">
                        <div class="container">
                            <!--Row with two equal columns-->
                            <div class="row">
                                <div class="col">This task</div>
                                <div class="col"><button type="button" class="btn btn-primary"
style="align-items: left;">View</button></div>
                            </div>
                        </div>
                    </div>
                    <p>task 2</p>
                    <p>task 3</p>
                </div>
            </div>
            <div class="col-xl-4">
                <div class="container text-white border py-3 my-3" style="background-color:  rgb(1, 1,
129);">
                    <h2>Previous tasks</h2>
                    <p>task 1</p>
                    <p>task 2</p>
                    <p>task 3</p>
                </div>
            </div>
            <div class="col-xl-4">
                <div class="container text-white border py-3 my-3" style="background-color:  rgb(1, 1,
129);">
                    <h2>Create a task</h2>
                    <p><a href="create_task.html"><button type="button" class="btn btn-
primary">Create</button></a></p>
                </div>
            </div>
        </div>
    </div>

    <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

    <!-- Option 2: Separate Popper and Bootstrap JS -->
    <!--
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"
integrity="sha384-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"
crossorigin="anonymous"></script>
    -->
  </body>
</html>
```

Code Progression Itemization

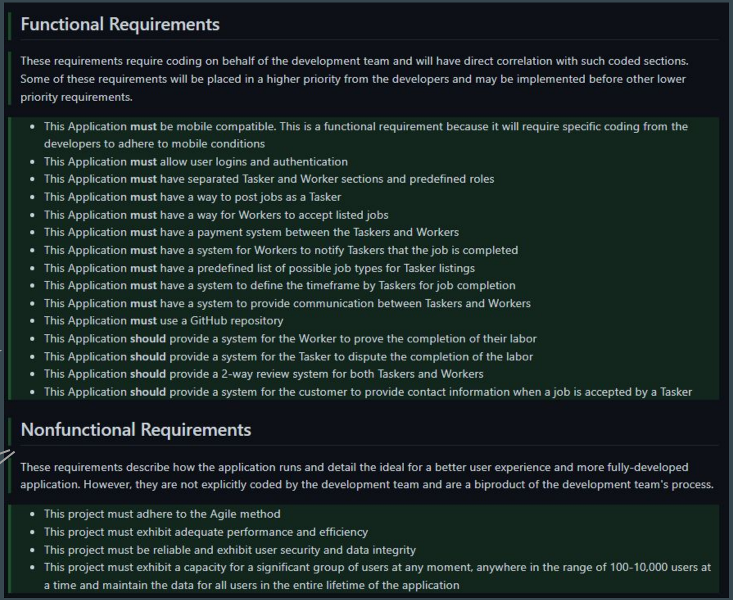


Type	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created ↓	Updated	Due
🟩	YOKE-23	Create controller class/ view for taskWorker child class	Jasper Swensen	Jasper Swensen	=	DONE	Done	Feb 24, 2022	Mar 24, 2022	
🟩	YOKE-22	Create controller classes/view for taskPoster child class	Jasper Swensen	Jasper Swensen	=	DONE	Done	Feb 24, 2022	Mar 24, 2022	
🟩	YOKE-21	Create Activity Diagrams	MC Michael C	MC Michael C	=	DONE	Done	Feb 9, 2022	Feb 14, 2022	
🟩	YOKE-20	Update and revise use case diagrams for milestone 2	BM Brian Mathews	Jasper Swensen	=	DONE	Done	Feb 3, 2022	Feb 14, 2022	
🟩	YOKE-19	Revise requirements from milestone 1 to include any additional stuff	BM Brian Mathews	MC Michael C	︾	DONE	Done	Feb 3, 2022	Feb 14, 2022	
🟩	YOKE-18	Update and revise readme for milestone 2	BM Brian Mathews	Jasper Swensen	︾	DONE	Done	Feb 3, 2022	Feb 14, 2022	
🟩	YOKE-17	Update and revise project plan for milestone 2	BM Brian Mathews	Jasper Swensen	=	DONE	Done	Feb 3, 2022	Feb 14, 2022	
🟩	YOKE-16	Create Account info page template	BM Brian Mathews	MC Michael C	=	DONE	Done	Feb 3, 2022	Apr 11, 2022	
🟩	YOKE-15	Create Available tasks page template	Jasper Swensen	Jasper Swensen	=	DONE	Done	Feb 3, 2022	Mar 24, 2022	
🟩	YOKE-14	Create Login page template	BM Brian Mathews	Jasper Swensen	=	DONE	Done	Feb 3, 2022	Mar 24, 2022	
🟩	YOKE-13	Create controller classes/view for Review class	Jasper Swensen	Jasper Swensen	=	DONE	Done	Feb 3, 2022	Mar 24, 2022	
🟩	YOKE-12	Create controller classes/view for Task class	MC Michael C	MC Michael C	=	DONE	Done	Feb 3, 2022	Mar 24, 2022	
🟩	YOKE-11	Create controller classes/view for User superclass	Jasper Swensen	Jasper Swensen	=	DONE	Done	Feb 3, 2022	Mar 24, 2022	

Requirements Progression



```
## Functional Requirements
These requirements are necessary for the application to function on a minimum level and are dictated as such by the application dev
- This Application **must** be mobile compatible
- This Application **must** allow user logins and authentication
- This Application **must** have separated Tasker and Worker sections and predefined roles
- This Application **must** have a way to post jobs as a Tasker
- This Application **must** have a way for Workers to accept listed jobs
- This Application **must** have a payment system between the Taskers and Workers
- This Application **must** have a system for Workers to notify Taskers that the job is completed
- This Application **must** have a predefined list of possible job types for Tasker listings
- This Application **must** have a system to define the timeframe by Taskers for job completion
- This Application **must** have a system to provide communication between Taskers and Workers

## Nonfunctional Requirements
These requirements are not necessary for the application to function on a minimum level but are ideal for a better user experience
- This Application **should** provide a system for the Worker to prove the completion of their labor
- This Application **should** provide a system for the Tasker to dispute the completion of the labor
- This Application **should** provide a 2-way review system for both Taskers and Workers
- This Application **should** provide a system for the customer to provide contact information when a job is accepted by a Tasker
```



Functional Requirements

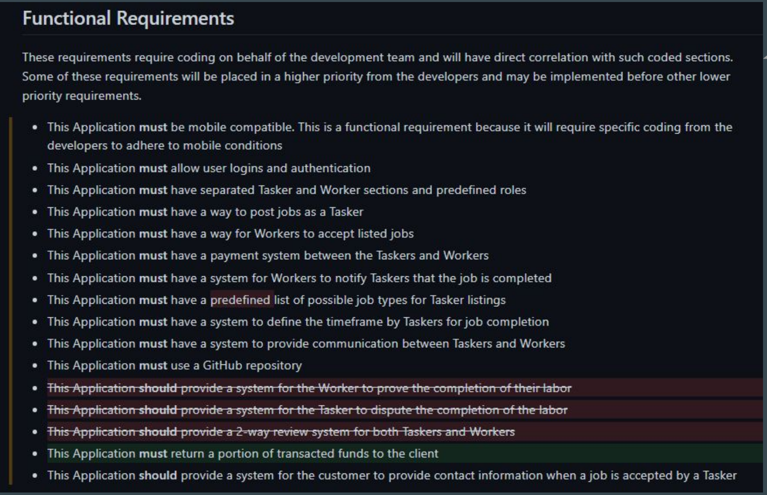
These requirements require coding on behalf of the development team and will have direct correlation with such coded sections. Some of these requirements will be placed in a higher priority from the developers and may be implemented before other lower priority requirements.

- This Application **must** be mobile compatible. This is a functional requirement because it will require specific coding from the developers to adhere to mobile conditions
- This Application **must** allow user logins and authentication
- This Application **must** have separated Tasker and Worker sections and predefined roles
- This Application **must** have a way to post jobs as a Tasker
- This Application **must** have a way for Workers to accept listed jobs
- This Application **must** have a payment system between the Taskers and Workers
- This Application **must** have a system for Workers to notify Taskers that the job is completed
- This Application **must** have a predefined list of possible job types for Tasker listings
- This Application **must** have a system to define the timeframe by Taskers for job completion
- This Application **must** have a system to provide communication between Taskers and Workers
- This Application **must** use a GitHub repository
- This Application **should** provide a system for the Worker to prove the completion of their labor
- This Application **should** provide a system for the Tasker to dispute the completion of the labor
- This Application **should** provide a 2-way review system for both Taskers and Workers
- This Application **should** provide a system for the customer to provide contact information when a job is accepted by a Tasker

Nonfunctional Requirements

These requirements describe how the application runs and detail the ideal for a better user experience and more fully-developed application. However, they are not explicitly coded by the development team and are a biproduct of the development team's process.

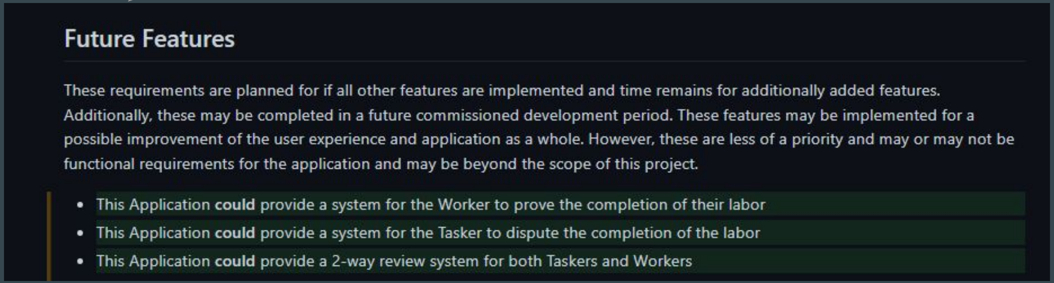
- This project must adhere to the Agile method
- This project must exhibit adequate performance and efficiency
- This project must be reliable and exhibit user security and data integrity
- This project must exhibit a capacity for a significant group of users at any moment, anywhere in the range of 100-10,000 users at a time and maintain the data for all users in the entire lifetime of the application



Functional Requirements

These requirements require coding on behalf of the development team and will have direct correlation with such coded sections. Some of these requirements will be placed in a higher priority from the developers and may be implemented before other lower priority requirements.

- This Application **must** be mobile compatible. This is a functional requirement because it will require specific coding from the developers to adhere to mobile conditions
- This Application **must** allow user logins and authentication
- This Application **must** have separated Tasker and Worker sections and predefined roles
- This Application **must** have a way to post jobs as a Tasker
- This Application **must** have a way for Workers to accept listed jobs
- This Application **must** have a payment system between the Taskers and Workers
- This Application **must** have a system for Workers to notify Taskers that the job is completed
- This Application **must** have a predefined list of possible job types for Tasker listings
- This Application **must** have a system to define the timeframe by Taskers for job completion
- This Application **must** have a system to provide communication between Taskers and Workers
- This Application **must** use a GitHub repository
- ~~This Application **should** provide a system for the Worker to prove the completion of their labor~~
- ~~This Application **should** provide a system for the Tasker to dispute the completion of the labor~~
- ~~This Application **should** provide a 2-way review system for both Taskers and Workers~~
- This Application **must** return a portion of transacted funds to the client
- This Application **should** provide a system for the customer to provide contact information when a job is accepted by a Tasker

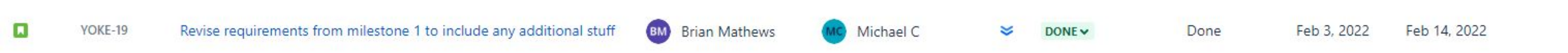


Future Features

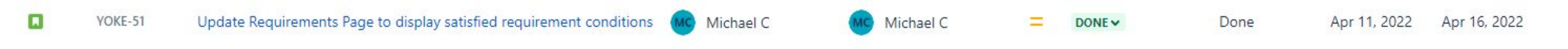
These requirements are planned for if all other features are implemented and time remains for additionally added features. Additionally, these may be completed in a future commissioned development period. These features may be implemented for a possible improvement of the user experience and application as a whole. However, these are less of a priority and may or may not be functional requirements for the application and may be beyond the scope of this project.

- This Application **could** provide a system for the Worker to prove the completion of their labor
- This Application **could** provide a system for the Tasker to dispute the completion of the labor
- This Application **could** provide a 2-way review system for both Taskers and Workers

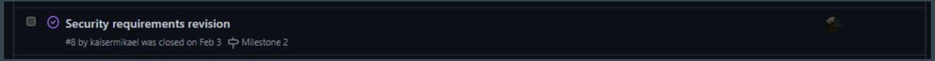
Requirements Progression Itemization



🟩	YOKE-19	Revise requirements from milestone 1 to include any additional stuff	BM Brian Mathews	MC Michael C	︾	DONE ⌄	Done	Feb 3, 2022	Feb 14, 2022

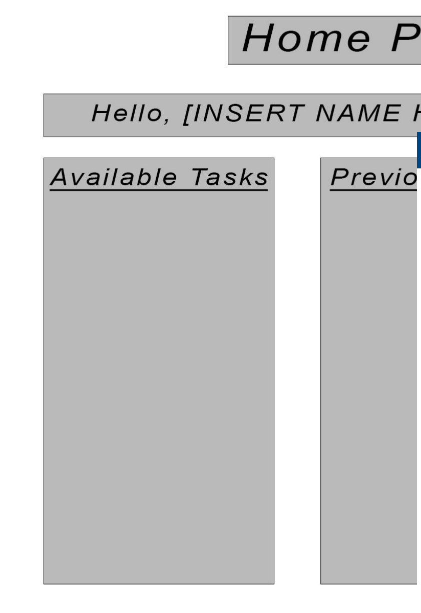


🟩	YOKE-51	Update Requirements Page to display satisfied requirement conditions	MC Michael C	MC Michael C	=	DONE ⌄	Done	Apr 11, 2022	Apr 16, 2022



□ ✓ **Security requirements revision**
#8 by kaisermikael was closed on Feb 3 Milestone 2

Prototype Progression



Home Page

Account

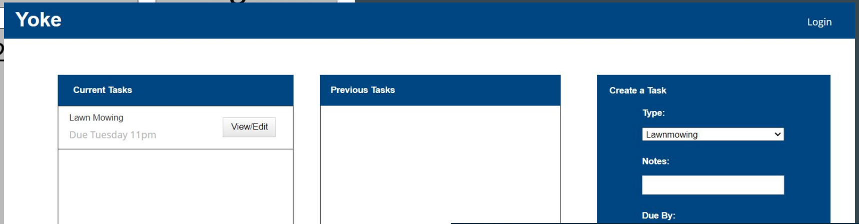
Hello, [INSERT NAME HERE]

Logout

Available Tasks

Previo

Low Fidelity



Yoke

Login

Current Tasks

Lawn Mowing

Due Tuesday 11pm

View/Edit

Previous Tasks

Create a Task

Type:

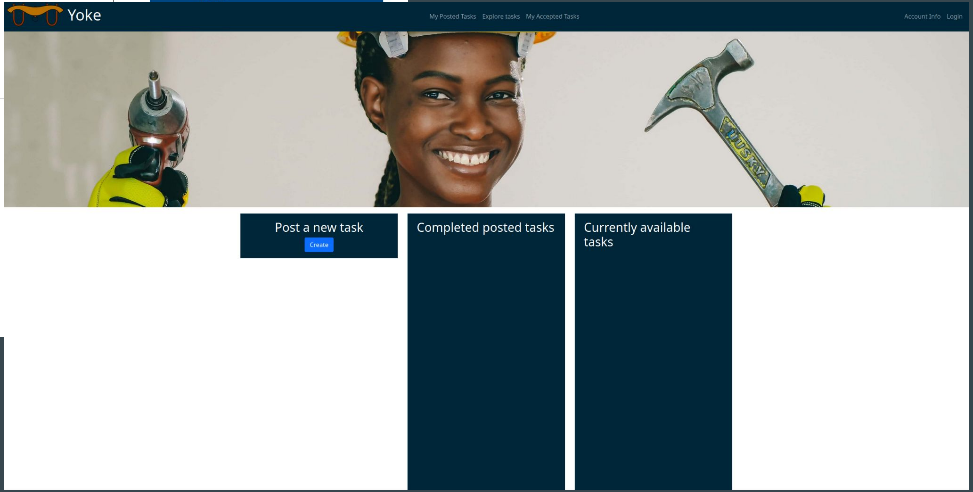
Lawnmowing

Notes:

Due By:

High Fidelity

Final Product



Yoke

My Posted Tasks

Explore tasks

My Accepted Tasks

Account Info

Login

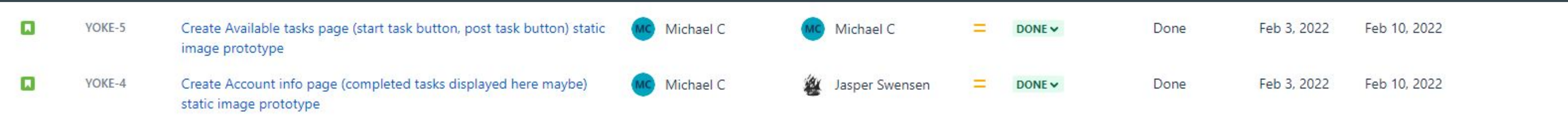
Post a new task

Create

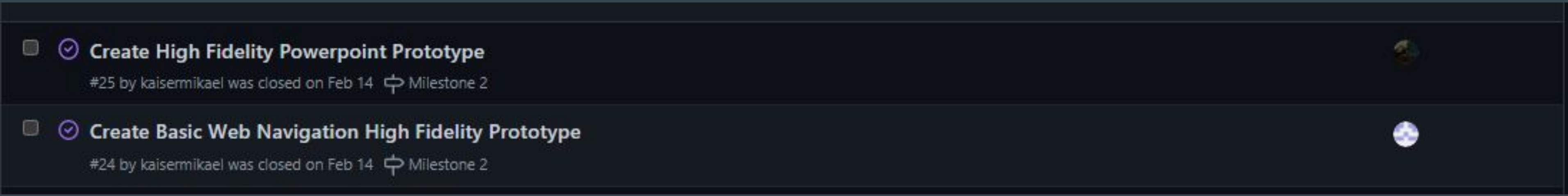
Completed posted tasks

Currently available tasks

Prototype Progression Itemization



YOKE-5	Create Available tasks page (start task button, post task button) static image prototype	Michael C	Michael C	=	DONE	Done	Feb 3, 2022	Feb 10, 2022
YOKE-4	Create Account info page (completed tasks displayed here maybe) static image prototype	Michael C	Jasper Swensen	=	DONE	Done	Feb 3, 2022	Feb 10, 2022



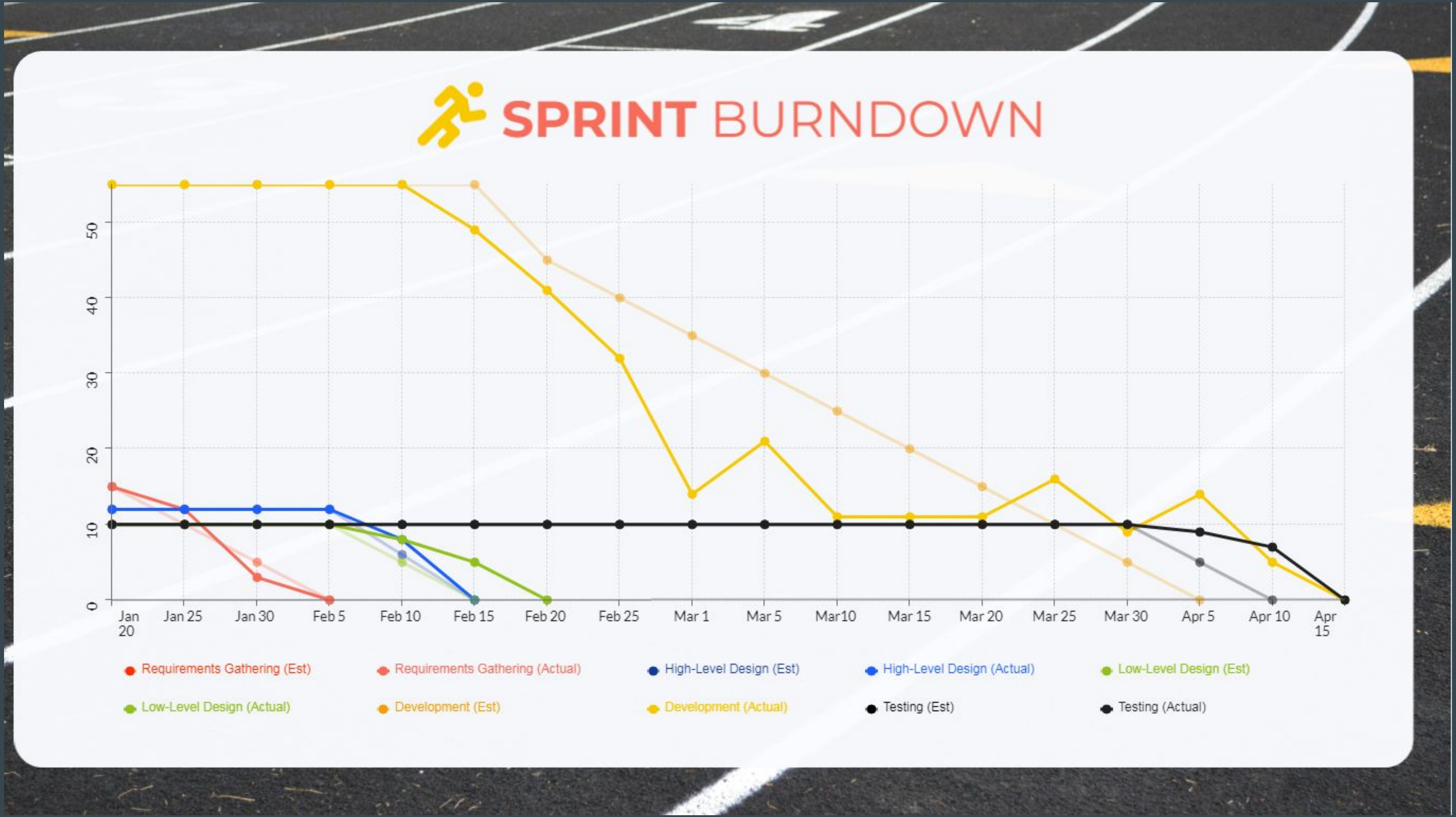
□ **Create High Fidelity Powerpoint Prototype**

#25 by kaisermikael was closed on Feb 14 Milestone 2

□ **Create Basic Web Navigation High Fidelity Prototype**

#24 by kaisermikael was closed on Feb 14 Milestone 2

Project Burndown Chart



Resources

- *Django documentation*. Django. (n.d.). Retrieved February 15, 2022, from https://docs.djangoproject.com/en/4.0/
- Mark Otto, J. T. (n.d.). *Introduction to Bootstrap*. · Bootstrap. Retrieved March 3, 2022, from https://getbootstrap.com/docs/4.1/getting-started/introduction/
- *Django bootstrap: An essential framework to beat your competitors!* DataFlair. (2021, June 21). Retrieved February 26, 2022, from https://data-flair.training/blogs/django-bootstrap/
- Atlassian. (n.d.). *Jira: Issue & project tracking software*. Atlassian. Retrieved February 2, 2022, from https://www.atlassian.com/software/jira
- GitHub. (n.d.). Retrieved April 16, 2022, from https://github.com/
- Unsplash. (n.d.). *Beautiful free images & pictures*. Unsplash. Retrieved April 3, 2022, from https://unsplash.com/

Questions?