



# TagTeam Dashboard

Improving Team Collaboration

The Jar Files  
Jacob Morroni, Ali Manzoor,  
Rishi Singh, Nebiyu Abreha

# Original Problem Statement

## Client-based Projects

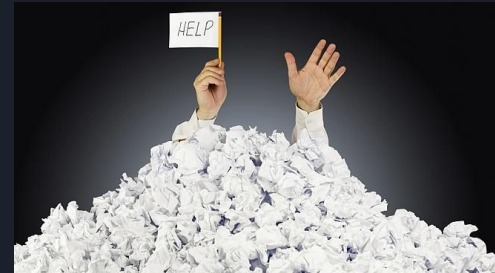
- Changing requirements
- Loose incentive to assist other teams

## Pre Deployment Stage

- Disorganized tasks
- Uneven workloads
- Difficult to request additional help

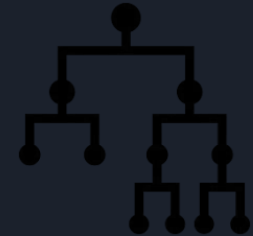
## Post Development Stage

- Cluttered issue tickets



# TagTeam Dashboard

- Tagging Feature: Identify and tag problematic code sections for peer assistance.
- GitHub Integration: Seamlessly sync code changes with GitHub.
- Reward System: Incentivize developers for helping peers.
- Task Assignment and Management: Efficiently distribute and track tasks.



# Tagging

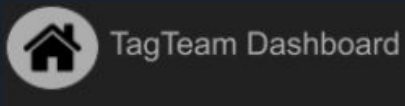
- What it does:
  - Tag specific sections of code to request help from team members.
  - Add descriptions to provide context for the issue.
  - Receive team notifications when a new tag is created.
- Why it matters:
  - Encourages direct and specific collaboration.
  - Speeds up problem resolution.





# Github Integration

- What It Does:
  - Edit, commit, and push code directly from the dashboard.
  - Real-time synchronization of task statuses with GitHub commits.
  - Notifications for code changes and updates.
- Why It Matters:
  - Reduces context-switching between tools.
  - Keeps project updates synchronized and transparent.



# Reward System



- What It Does:

- Tracks contributions for resolving tagged issues.
- Awards points and displays leaderboards.
- Recognizes and rewards developers for collaboration.



- Why It Matters:

- Motivates teamwork and knowledge sharing.
- Creates a positive and collaborative environment.





# Task Assignment and Management

- What It Does:
  - Assign tasks to sub-teams with clear ownership.
  - Track task progress in real time on a task board.
  - Prevents uneven workload distribution.
- Why It Matters:
  - Improves project organization and accountability.
  - Ensures smooth task flow and efficient team performance.




# Visual Representation

 TagTeam Dashboard



App.js (Version 1.3.2)  
(Assigned to UI/UX team)(13 Code Tags, 3 Unread Comments)



Discussion Board

Code Editor

GitHub Integration

John Smith

I feel like this line could be conde...

Alexis C.

Discussion Closed

Sean Gold


This export should be adjusted to ...


```
1 import React, { Component } from "react";
2 import './App.css';
3 import Container from "react-bootstrap/Container";
4 import { BrowserRouter as Router } from "react-router-dom";
5
6 class App extends Component {
7   constructor(props) {
8     super(props);
9     this.state = {
10       title: "Aayush Bajracharya",
11       headerButtons: [
12         { title: "Home", path: "/" },
13         { title: "About", path: "/" },
14         { title: "Contact", path: "/" },
15       ],
16     };
17   }
18   state = {};
19   render() {
20     return (
21       <Router>
22         <Container>Hello World</Container>
23       </Router>
24     );
25   }
26 }
27
28 export default App;
```


Changes


History

☒ 3 changed files

☒ App.js 

☒ style.css 

☒ index.html 

 Summary (required)

Description

Commit to Test\_Branch



# Course Concepts Used and Their Utility

## 1. Agile

- Most applicable concept from class
- Used throughout entire project
- Increased efficiency and overall team awareness

## 2. Requirements Engineering

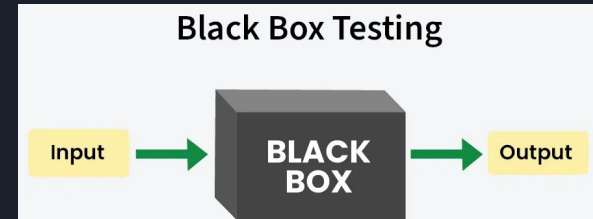
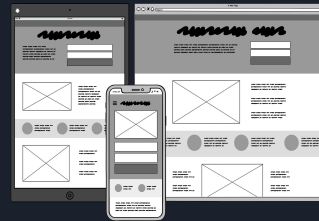
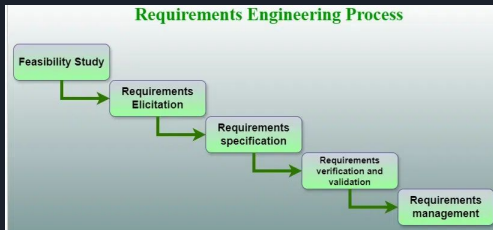
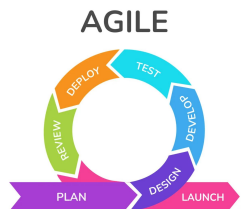
- Requirements elicitation was somewhat useful since external input helped shape our project
- Requirements analysis was, because it organized the team's goals for what the product would look like
- Requirements specification was not useful since

## 3. Wireframing

- Useful to visualize project after weeks of just working with words and concepts

## 4. Black Box Testing

- Not useful since code wasn't implemented but interesting to think about testing





# Limitations & Future Work

## Limitations:

- Real-time Notifications: Delays can occur in high-traffic environments or under system load.
- File Types: Limited support for file types (maybe only PDFs, docs, and images).

# Related Tools

1. Kanban Board for task tracking and collaboration.
2. Git for version control and code management.
3. Slack for team communication and announcements.





# Things We Learned

1. Learned how to break the project down into smaller tasks (sprints) for agile development
  - a. Prevented us from falling behind
  - b. Helped us create priorities and stay on task
2. Learned how to conduct stand-up meetings during class time
  - a. Allowed us to communicate progress on milestones
  - b. Allowed us to distribute work and discuss strategies to implement and test the product
3. Learned how to conduct the process of requirements elicitation
  - a. Allowed us to understand what developers would want in the product to better tailor it
4. Learned how to document progress of a project through project milestones
  - a. Allowed us to track our progress and create records to go back and review milestones