**Social Networking Cop**

*University of Maryland Global Campus*

*CMSC 495 6982 Capstone in Computer Science (2255)*

*Reid Buchanan*

*Lemar Llanes*

*David Jarrett*

*July 06, 2025*

**Table of Contents**

[***Phase I Project Overview and Requirements*** 1](#_Toc202618248)

[***Project Scope:*** 1](#_Toc202618249)

[***Objectives:*** 1](#_Toc202618250)

[***Phase I Deliverables:*** 1](#_Toc202618251)

[***Success Criteria:*** 2](#_Toc202618252)

[***Software Architecture Overview*** 2](#_Toc202618253)

[***Component Interactions:*** 3](#_Toc202618254)

[***Implementation Note:*** 3](#_Toc202618255)

[***Software Implementation*** 4](#_Toc202618256)

[***Coding Process:*** 4](#_Toc202618257)

[***Testing and Debugging:*** 5](#_Toc202618258)

[***Testing Approach:*** 5](#_Toc202618259)

[***Types of Testing Conducted:*** 5](#_Toc202618260)

[***Test Results Summary:*** 6](#_Toc202618261)

[***Feed.HTML – Source Code:*** 6](#_Toc202618262)

[***Conclusion*** 9](#_Toc202618263)

## ***Phase I Project Overview and Requirements***

### ***Project Scope:***

The Social Networking Cop project is designed to promote healthier screen time habits by monitoring and alerting users when they exceed predefined time limits on a faux social networking platform called TimeSpace. The application tracks cumulative daily and weekly screen time and notifies users when thresholds are met or exceeded.

Phase I focuses on building the foundational components of this system, including the user interface and the initial time-tracking logic.

### ***Objectives:***

The objectives of Phase I are:

* Establish a working project environment with tools and workflows.
* Develop the basic HTML structure for the login and social feed pages.
* Implement core JavaScript functionality to track screen time after login.
* Display alert prompts when daily (20 minutes) or weekly (2 hours) screen time limits are exceeded.
* Validate the accuracy of time tracking through preliminary unit testing.
* Document the code and system architecture for future phases.

### ***Phase I Deliverables:***

The key deliverables for Phase I are:

* index.html (login interface)
* feed.html (basic social media feed interface)
* app.js (core JavaScript time-tracking script)
* Preliminary unit tests for time-tracking logic
* Inline code documentation and project README file
* High-level software architecture description
* Phase I Source Code Document (this document)

### ***Success Criteria:***

Phase I will be considered a success if:

* Users can log in and access the social feed through the HTML pages.
* The JavaScript timer accurately tracks screen time from login onward.
* Alert prompts are displayed when the daily and weekly time limits are exceeded.
* Time data is temporarily stored using local storage.
* Preliminary unit tests confirm time tracking and alert functionalities work as intended.
* All components are documented with inline comments and a README file created.
* The software architecture and system component interactions are clearly defined.

## ***Software Architecture Overview***

The software architecture for the Social Networking Cop project defines the main components of the system, their responsibilities, and how they interact to meet the project’s requirements. This high-level design outlines the foundational structure implemented in Phase I.

**System Components:**

**HTML user interface (UI):**

* Provides the structure for user login and faux social feed.
* Pages include index.html and feed.html.

**JavaScript Time Tracking Logic:**

* Implemented in app.js.
* Begins tracking when a user logs in.
* Monitors cumulative daily and weekly usage.
* Alert mechanisms to prompt the user when limits are exceeded:
  + 20 minutes/day
  + 2 hours/week

**Local Storage:**

* Used to persist daily and weekly screen time data across page views within a single browser session.

### ***Component Interactions:***

When a user logs into TimeSpace through index.html, the JavaScript app.js script initiates and begins tracking time. As the user moves to feed.html, the script continues monitoring and updating time in local storage. If the cumulative time exceeds 20 minutes/day or 2 hours/week, the system triggers alert messages. The UI and background logic communicate via DOM events and shared local storage.

### ***Implementation Note:***

This Phase I implementation uses client-side HTML and JavaScript only. All code run locally in the user’s browser, with data persisted using local storage. No server-side components, data, or third-party libraries are integrated at this stage.

## ***Software Implementation***

The software implementation section describes how the software system’s core components were developed during Phase 1. It covers the programming languages used, the files created, and the coding process, including testing and debugging activities.

Implementation Overview:

In Phase 1, the team developed the foundational components of the Social Networking Cop project using HTML and JavaScript. The implementation included:

* Index.html – Login Page
* Feed.html – Faux Social Feed Page
* App.js – JavaScript Time-Tracking Script

All components were created using Visual Studio Code as the primary development environment, with testing conducted via Safari Developer Tools and local browser sessions.

### ***Coding Process:***

* Created the HTML structure for login and feed pages, including form elements and navigation links.
* Developed the JavaScript app.js file to:
  + Start a timer when the user logs in
  + Track Cumulative daily and weekly screen time using local storage
  + Trigger alert prompts when time limits are exceeded
* Added inline comments throughout the code for clarity and maintainability.

### ***Testing and Debugging:***

* Conducted preliminary unit tests on the JavaScript timer functionality to validate accurate tracking.
* Verified that alert prompts display when daily and weekly thresholds are met.
* Debugged issues related to timer reset behavior and local storage persistence.
* Logged test outcomes in development notes.

**Software Testing**

The Software Testing section documents how the team verified that the software components met the Phase I requirements, as specified in the Project Plan and Project Design. It covers unit testing, integration testing, and preliminary system testing efforts.

### ***Testing Approach:***

Testing was performed manually using black-box testing methods during Phase I, focusing on verifying:

* Accurate time tracking after login.
* Correct Triggering of alert prompts
* Data persistence within local storage

### ***Types of Testing Conducted:***

|  |  |  |
| --- | --- | --- |
| Testing Type | Description | Status |
| Unit Testing | Tested individual JavaScript functions for timer control and alert logic. | Completed |
| Integration Testing | Verified that index.html, feed.html, and timer.js integrated and functioned together without errors. | Completed |
| System Testing | Conducted basic end-to-end test to confirm users could log in, use the platform, and receive alerts when limits were exceeded. | Completed |

### ***Test Results Summary:***

* Timer accurately tracked session duration.
* Daily and weekly alert thresholds functioned correctly.
* Local storage successfully stored and retrieved time data.
* No major defects identified in core Phase I functionalities.

Note: Additional agile approaches are planned for Phase II.

### ***Feed.HTML – Source Code:***

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>TimeSpace Feed</title>

  <link rel="stylesheet" href="style.css">

  <!-- If JavaScript is disabled, auto-redirect to login -->

  <noscript>

    <meta http-equiv="refresh" content="0;url=index.html">

  </noscript>

  <style>

    .top-bar {

      display: flex;

      justify-content: flex-end;

      align-items: center;

      padding: 10px 20px;

      background-color: navy;

      color: white;

      flex-wrap: wrap;

    }

    .top-bar button, .top-bar select {

      margin-left: 10px;

      padding: 8px 16px;

      border: none;

      border-radius: 6px;

      cursor: pointer;

      font-weight: bold;

    }

    .sign-out-btn { background-color: crimson; }

    .add-user-btn { background-color: seagreen; color: white; }

    .view-users-btn { background-color: #4444aa; color: white; }

    .feed-container {

      padding: 20px;

      max-width: 600px;

      margin: 0 auto;

    }

    .post-box {

      display: flex;

      flex-direction: column;

      margin-bottom: 20px;

    }

    .post-box textarea {

      padding: 10px;

      border: 1px solid #ccc;

      border-radius: 6px;

      resize: vertical;

      margin-bottom: 10px;

      min-height: 80px;

    }

    .post-box button {

      align-self: flex-end;

      background-color: navy;

      color: white;

    }

    .post {

      background-color: #fff;

      padding: 15px;

      border-radius: 8px;

      box-shadow: 0 2px 8px rgba(0,0,0,0.1);

      margin-bottom: 15px;

    }

    .post .meta {

      font-size: 12px;

      color: #555;

      margin-bottom: 8px;

    }

    .screen-time {

      margin: 20px 0;

      text-align: center;

      font-size: 16px;

      color: green;

    }

  </style>

</head>

<body>

  <div class="top-bar">

    <span id="usernameDisplay"></span>

    <button class="add-user-btn" onclick="addUser()">Add User</button>

    <button class="view-users-btn" onclick="viewUsers()">View Users</button>

    <select id="userDropdown" onchange="changeUser()" style="padding: 6px; border-radius: 6px;"></select>

    <button class="sign-out-btn" id="signOutBtn">Sign Out</button>

  </div>

  <header>

    <h1>TimeSpace Feed 📣</h1>

  </header>

  <div class="feed-container">

    <div class="screen-time" id="screenTime"></div>

    <div class="post-box">

      <textarea id="postContent" placeholder="What's on your mind?"></textarea>

      <button onclick="addPost()">Post</button>

    </div>

    <div id="feed">

      <div class="post">

        <div class="meta">TimeSpace Bot · Just now</div>

        <p>Welcome to the TimeSpace community! 🎉 Drop your first post above.</p>

      </div>

    </div>

  </div>

  <footer>

    <p>&copy; TimeSpace</p>

  </footer>

  <!-- Inline redirect check in case app.js fails -->

  <script>

    // Quick redirect check for username and startTime presence

    (function() {

      const username = localStorage.getItem("username");

      const startTime = localStorage.getItem("startTime");

      if (!username || !startTime) {

        window.location.href = "index.html";

      }

    })();

  </script>

  <!-- Main application script -->

  <script src="app.js" defer></script>

  <script>

    window.onload = () => {

      startFeedTimer();

      document.getElementById("signOutBtn").onclick = signOut;

      const username = localStorage.getItem("username");

      document.getElementById("usernameDisplay").textContent = username ? `Signed in as ${username}` : "";

      updateUserDropdown();

    };

  </script>

</body>

</html>

Due to document length limitations, only the feed.html source code is included in the Phase I submission. The supporting index.html and app.js files have been developed, tested, and integrated into the system, and are available upon request.

## ***Conclusion***

The Phase I source code establishes the foundation for the Social Networking Cop project, demonstrating core skills in software design, implementation, and testing. It meets the initial requirements outlined in the project plan and prepares the system for integration and further development in the phase.