**Link Tailor**

Team Contract

University of Cincinnati

College of Education, Criminal Justice and Human Services

School of Information Technology

Senior Design Project 2021

Daniel Hickman

Kemal Ozturk

Lauren Tillery

## Table of Contents

[Table of Contents 2](#_Toc49282802)

[Contract Intent 3](#_Toc49282803)

[The Team 4](#_Toc49282804)

[Team Values & Rules 6](#_Toc49282805)

[Communication Plan 7](#_Toc49282806)

[Problems and Escalations 7](#_Toc49282807)

[The Project 8](#_Toc49282808)

[The Problem 8](#_Toc49282809)

[The Solution 8](#_Toc49282810)

[Project Source 8](#_Toc49282811)

[Project Goals 9](#_Toc49282812)

[Project Scope 10](#_Toc49282813)

[Project Timeline 10](#_Toc49282814)

[Technologies Used 10](#_Toc49282815)

[Team Signatures 11](#_Toc49282816)

## Contract Intent

The following contract was written and agreed upon by Daniel Hickman, Kemal Ozturk, and Lauren Tillery. The contract provides expectations, objectives, and methods for the development of the Link Tailor Application.

This contract is effective for all team members participating in IT 5001/5002 during the 2020-2021 school year.

# The Team

“The team” refers to the three parties of this contract who are working together to create the Link Tailor application.

**Daniel Hickman**

Project Manager

leinadhickman@gmail.com

hickmadc@mail.uc.edu

513-438-1597

Responsibilities

* Schedule and facilitate team meetings based on the Agile methodology. This will include Sprint Planning Sessions, Stand-ups, Sprint Reviews, and Retrospectives.
* Track velocity and update the project timeline.
* Consider future features and groom the Product Backlog with the team.
* Participate in the development work.
* Present completed work at the Sprint Reviews and to class advisors.

**Kemal Ozturk**

Developer

97kemalozturk@gmail.com

ozturkkl@mail.uc.edu

832-520-1467

Responsibilities

* Develop code for the Link Tailor Application.
* Self-assign and create Kanban tasks to reflect work in-progress and completed.
* Present completed work at the Sprint Reviews and to class advisors.
* Estimate future work in the Sprint Planning sessions.
* Research features and technology that could improve the Link Tailor Application.

**Lauren Tillery**

Developer

tillerylauren@gmail.com

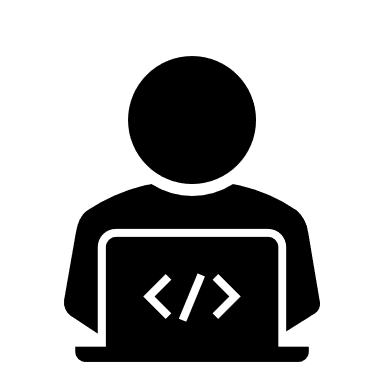
tillerlo@mail.uc.edu

513-240-6907

Responsibilities

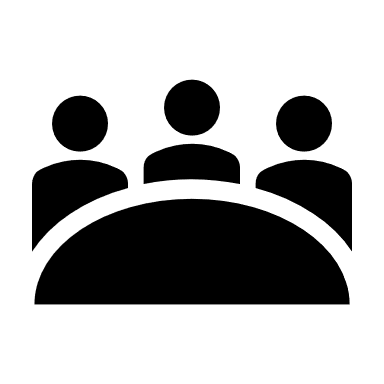
* Develop code for the Link Tailor Application.
* Self-assign and create Kanban tasks to reflect work in-progress and completed.
* Present completed work at the Sprint Reviews and to class advisors.
* Estimate future work in the Sprint Planning sessions.
* Research features and technology that could improve the Link Tailor Application.

## Team Values & Rules



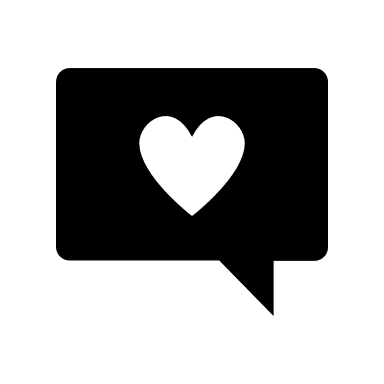
**Working Software**

Working hard to create a satisfying product.



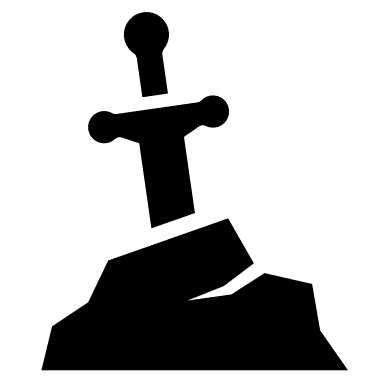
**Communication**

Frequent, effective, and honest communication.



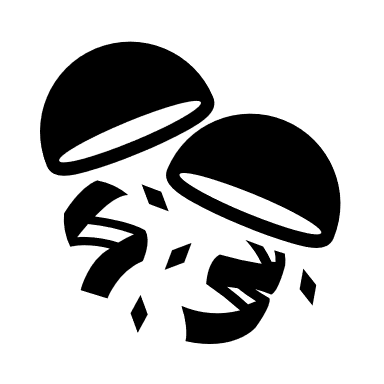
**Empathy**

Assuming the best of one another, treating each other with respect and care.



**Integrity**

Doing what is right, avoiding deceit, and owning up to mistakes.



**Celebrating the Try**

Our efforts are worthy of

recognition and celebration!

**Additional Rules**

1. Plagiarism will not be tolerated. Any team member that plagiarizes will be subject to university policies and a team meeting will be called.
2. Each team member will stay current on their tasks to ensure the project milestones are being met. If an event conflicts that will affect the completion of a deliverable, the team member will notify the other team members at least 24 hours in advance of the scheduled due date.
3. If a group member will be absent on class days or for an extended period of time, they will notify the other team members and the Instructors.
4. All team members are required to attend all scheduled meetings and provide updates to the acting project manager during the meeting. If a team member cannot make a scheduled meeting, they must notify all team members at least 4 hours in advance and provide an update via messenger or email.
5. All team members will review the oral presentation and final white paper.
6. All team members will respect the opinions and ideas of each team member, other students, and faculty.

## Communication Plan

**Meetings**

We plan to have 2-week long Sprints with the following recurring meetings:

10 minute **Stand-Up** meetings every Monday, Wednesday, and Friday.

* Concerns/impediments?
* Updates to share?

1 hour **Sprint Planning** meetings every Monday at the beginning of a new Sprint.

* Assign Story Points to tasks
* Determine acceptance criteria for each assigned task

Half-hour **Sprint Reviews** every Friday at the end of a Sprint.

* Review each completed task based on the acceptance criteria

Half-hour **Sprint Retrospectives** every Friday at the end of a Sprint.

**Kanban Board**

We plan to use a Kanban Board through GitHub Projects. It is expected that all team members use the board to update their work. The Kanban tasks will likely become the primary place for communicating feature and project requirements and acceptance criteria of work.

**Code**

In this project, it is always preferable to over-comment rather than under-comment. As a team of relatively inexperienced developers, we need all the help we can get from frequent, verbose comments. If after the code is fully tested, we decide some comments are unnecessary clutter and want to trim them up we can.

## Problems and Escalations

If a team member has a concern with another member, they must express their concern to that member. If the problem persists, all team members must discuss the issue with each other, in face-to-face or video discussion. If a resolution still cannot be reached, members of the team may then reach out to their Senior Design advisors/professors for advice and mediation.

# The Project

The project is the creation of a desktop application that functions as a hub for link management – one place from which to organize and open all regularly visited web links, device apps, and system files.

## The Problem

Keeping track of browser bookmarks, desktop icons, file system shortcuts and other “quick links” can be a frustrating task for computer users who use many links a day and frequently add new ones. Desktops get cluttered and bookmarks bars fill up. The current options for managing links (including Chrome bookmarks and the Windows start menu) have been found lacking in functionality and convenience. For instance, Chrome bookmarks is visually cramped and does not allow for spatial organization while the Windows start menu limits the ability to add web links or link to files directly. Users waste time when they hunt for links, whether it is the website they bookmarked earlier that day or a document they downloaded a week ago. This is an increasingly relevant concern as a growing number of computer users[[1]](#endnote-1) are accessing a growing number of websites and applications on a daily basis[[2]](#endnote-2).

## The Solution

The Link Tailor application will provide a space for users to organize and style links to their preference, empowering them to create a more enjoyable, personalized user-experience on their device. With frequently used links consolidated to one space and categorized according to the user’s preference, users can spend less time searching for their saved links, and more time doing what they want on their device. Users will be able to quickly access various layouts of links via hotkey, the links being organized spatially, with tags, and with style theming. The layouts will be able to be saved and imported to other devices so users can use the same layout on multiple devices and share layouts with others.

## Project Source

This project was conceived by Daniel Hickman, who found himself suffering from an over-abundance of bookmarks: more than 300, sorted into various folders such as “Games,” “Technology,” “Reading,” “Podcasts,” “Art/Music,” “Work,” “Recipes,” “D&D,” and “Return To.” Finding and organizing saved bookmarks was becoming an increasingly difficult and frustrating task. In investigating possible solutions such as Tagpacker, Pocket, Dewey bookmarks, Dragdis, Dropmark, and Station, all were found significantly lacking. Daniel did end up using RocketDock and the Windows Start menu for better organization of file-system links and applications, but neither provided any improvement to the original concern of bookmark management.

Thus, the concept for an easy way to organize not only bookmarks, but also file system files/folders and application links was born! Daniel found two other Software Development majors who were passionate about the project idea, Kemal and Lauren, and together they brainstormed ways to create an incredible app, using the exciting new technologies of Vue.js and Electron.

## Project Goals

**Functional**: Links can be added and organized.

* Web-links, files, folders, and application links can all be added and arranged spatially.
* Next-level functionality is “open with…” features where links can be set to run a particular program, as admin, or with other specifications.

**Convenience**: Quick to use.

* Measured by how many clicks/key-presses it takes to perform an action (such as adding a link). We want to minimize the steps it takes!
* This may mean creating browser extensions or Windows-integrations.

**Aesthetic**: The app looks good.

* Measured by user testing. Users will look at their layouts a lot so we want them to look awesome!

**Freedom**: Users are able to customize their experience.

* Measured by users being able to change the look and feel of their layouts easily. This might be done through allowing many variables to be changed by users very easily. This may also be done through providing more advanced users with the ability to craft and publish various designs.

**Sharing**: The ability to share design themes and link layouts with others.

* Measured by how easy and intuitive it is to export/import layouts.

**Website**: A space for users to share their layouts and download what others have created.

* Will promote community innovation around the app.
* If successful, new users will be able to browse and download beautiful themes and layouts.

**Data Recovery**: If a user’s PC is irrecoverable, they can load their link layouts on their new device.

* The first level is allowing for the importing and exporting of layout files within our app.
* The second level is creating an auto-back-up feature.

**Device Syncing**: A user’s change to a layout on one device will update a synced layout on another device.

## Project Scope

We, in using an Agile approach, will create user stories, assign story points to them, track our team’s velocity, and adjust our scope accordingly. Below are a list of features that are currently within scope, roughly organized in high-to-low priority.

**App Features**

* Working Application with basic functionality of the following features:
  + Links (add/edit/remove)
  + Drag-and-drop spatial-arrangement
  + Views
  + Tags
  + Searching
  + Expand-Boxes (groups)
* Extra options for opening links (admin, open with…)
* Themes & detailed customizations (color/shape/size)
* Custom hotkeys
* Importing/Exporting link layouts
* Chrome Extension
* Windows-integration (adding “New link…” to the right-click menu)
* Detailed searching filters
* An easy way for others to contribute themes, customizations, etc (& share them with others)
* Custom notifications, rotating messages, reminders, and triggers
* Syncing between devices
* Mobile Functionality (beginning with Android)
* Taskbar or dock/widget

## Project Timeline

---

## Technologies Used

**NPM**: Node environment has countess packages to support fully built applications ranging from file operations to user interface libraries. We will be able to utilize node package manager in our application.

**Electron**: Electron is an open-source framework that enables creating cross-platform applications using web technologies. It uses Chromium engine for the UI along with Node.js runtime for the back-end.

**JavaScript**: JavaScript programming language will be used to develop the application through node and electron.

**HTML**: Since the app is using Chromium engine for its front end. HTML will be used to map out the UI elements.

**CSS**: The styling of the UI elements will be set using Cascading Style Sheets.

**Vue.js**: Vue will be used to create the front-end as components to improve modularity and performance.

**SQLite**: For the storage and back-end database needs SQLite will be used. This is a great database solution that works amazing with Electron.

**Visual Studio Code**: A popular code editor made by Microsoft.

## Team Signatures

Signature: \_\_\_\_\_\_\_\_Daniel Hickman\_\_\_\_\_\_\_\_\_\_ Date: 8/25/2020

**Daniel Hickman**  Project Manager

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 8/25/2020

**Kemal Ozturk**  Developer

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 8/25/2020

**Lauren Tillery**  Developer

1. Alsop, Thomas. (2020). Computer penetration rate among households worldwide 2005-2019. Source/Publisher: ITU. Release Date: November 2019 <https://www.statista.com/statistics/748551/worldwide-households-with-computer/> [↑](#endnote-ref-1)
2. Statista Research Department. (2015) Number of websites visited by new and established internet users in the UK 2014. <https://www.statista.com/statistics/322946/number-of-websites-visited-by-new-and-established-internet-users/> [↑](#endnote-ref-2)