MICHIGAN STATE
U N I V E R S I T Y

Project Plan
AutoHook Editor

The Capstone Experience

Team Urban Science

Jon Stover
Ben LaFleur
Daiwei Zhang
Jeff Fallon
Zach Lewis

Department of Computer Science and Engineering
Michigan State University

Fall 2019

From Students…
…to Professionals

Functional Specifications

- AutoHook's central services must go through a long, inefficient process to make HTML changes for their client's AutoHook templates.

- They are looking for an efficient way to create, modify, and export their web designs.

- Our project core functionality allows them to modify websites in browser by bringing everything they need into one convenient application.

Design Specifications

- Landing page displaying their list of clients, selecting a client displays various projects.

- Once a project is selected, the web editing environment is opened, displaying the template on the left half of the screen with the HTML code on the right.

- After making changes, export the template to a relational database to be imported as needed later.
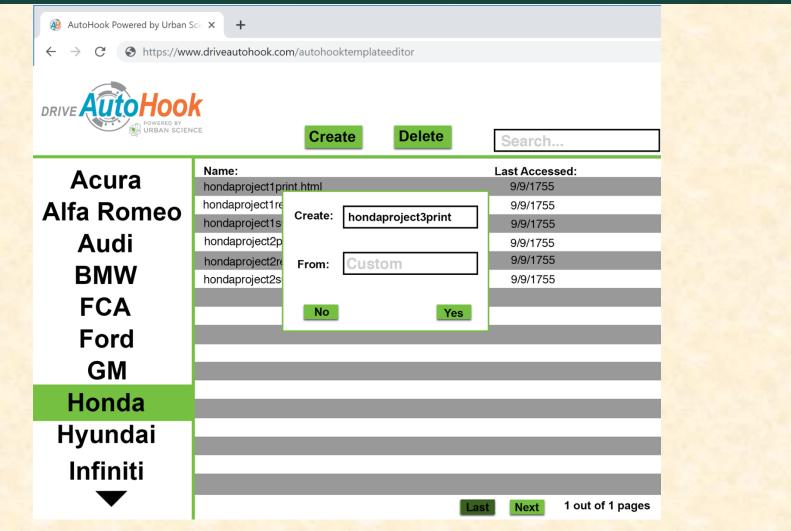
Screen Mockup: Project Select

Screen Mockup: Project Select Clicked

Screen Mockup: Project Select Create

Screen Mockup: Project Select Delete

Screen Mockup: Template Editor

Technical Specifications

- Our project is a data driven web application focused on modifying HTML through an intuitive UI

- Allows dynamic import/export of HTML/CSS template files using a relational database stored in the Microsoft Azure Cloud

- HTML editor displays live changes to code in iframe window

System Architecture

System Components

- Hardware Platforms
  - Microsoft Azure Cloud services
- Software Platforms / Technologies
  - Microsoft Visual Studio
    - ASP.net web development
    - JavaScript Angular framework front end
    - C# back end maintaining state information
  - Microsoft Azure Cloud
    - Relational database
    - Server

Risks

- Importing and Exporting HTML:
    - Description: Importing and exporting web page templates will require our project to move HTML to and from a relational database.
    - Mitigation: Once we get our web application to display data from our server, we will be experimenting with selecting and inserting large amounts of HTML and CSS from a database. We can then figure out how much of this content is necessary to select and insert, allowing us to minimize the need for storage on our server, and making importing/exporting web templates faster and more efficient.
- Display the HTML inside of Web App with Live Dynamic Updates:
    - Description: Our clients want a core aspect of our web application to be direct modification of the HTML and CSS that generates their content, from within a web browser.
    - Mitigation: We have researched an open source solution called ASP.net AJAX Control Toolkit which contains a tool called HTMLEditor that we believe we can modify to perform a good part of the html modification part of our project, allowing us to focus on our drag and drop tools.
- Intuitive GUI:
    - Description: Our client has a large volume of the templates that they wish for our tool to act on.
    - Mitigation: We have already built a relational database containing fields which have some basic html in them. We have performed a select statement on this database from within our application. We will wrap statements to create new items and manage files in buttons.
- Undo functionality:
    - Description: An intuitive interface should include the ability to undo actions.
    - Mitigation: We have a relational database which contains multiple copies of the html which serves up the template we are working on. We have an algorithm for having multiple versions which you can call back to.

Questions?