

# Nonprofit Condition Assessment – Version 2

Audrey Bunn, Kyle Dove, Justin Gill, Eve Neal, Kevin Todd, Steven Tucker

## North Carolina



Center *for* Nonprofits

# Table of Contents

<b>1.0 Introduction</b>	<b>5</b>
1.0.1 Project statement, Objections, and Functions	
<b>1.1 Project Estimates</b>	<b>6</b>
1.1.1 Historical Data	
1.1.2 Estimation Techniques	
1.1.3 Estimations	
<b>1.2 Project Risks</b>	<b>8</b>
<b>1.3 Schedule</b>	<b>9</b>
1.3.1 Project Work Breakdown Structure	
1.3.2 Task Network (CPM)	
1.3.3 Time-Line Chart (Gantt Chart)	
<b>1.4 Project Resources</b>	<b>12</b>
1.4.1 People	
1.4.2 Hardware	
1.4.3 Software	
<b>2.0 System Analysis</b>	<b>14</b>
2.0.1 System Requirements	
2.0.2 System Project Constraints	
2.0.3 System Reference Diagram	
<b>2.1 Behavioral Description</b>	<b>16</b>
2.1.1 Functional Partitioning	
2.1.2 Functional Description	
2.1.3 Control Description	
<b>2.2 Functional Description</b>	<b>17</b>
2.2.1 System States	
2.2.2 Events and Actions	

<b>2.3 Use Cases</b>	<b>19</b>
2.3.1 Stakeholders	
2.3.2 Actors	
2.3.3 Use Case Diagram	
2.3.4 Use Case Description	
<b>3.0 Scope</b>	<b>22</b>
3.0.1 System Objectives	
3.0.2 Major Software Requirements	
3.0.3 Design Constraints and Limitations	
<b>3.1 Data Design</b>	<b>23</b>
3.1.1 Data Objects	
3.1.2 Relationships	
<b>3.2 Architectural Design</b>	<b>25</b>
3.2.1 Review of Data and Control Flow	
3.2.2 Derived Program Structure	
<b>3.3 Interface Design</b>	<b>26</b>
3.3.1 Human-machine Interface Specification	
3.3.2 Human-machine Interface Design Rules	
3.3.3 External Interface Design	
3.3.4 Internal Interface Design	
<b>3.4 Procedural Design</b>	<b>30</b>
3.4.1 Processing Narrative	
3.4.2 Interface Description	
3.4.3 Design Language Description	
3.4.4 Modules Used	
3.4.5 Internal Data Structures	
3.4.6 Comments, Restrictions and Limitations	
<b>4.0 Test Plan</b>	<b>33</b>
4.0.1 What to Test	
4.0.2 When to Test	
4.0.3 How to Test	
<b>4.1 Functional Testing</b>	<b>34</b>

4.1.1 Test Data Design Technique(s)	
4.1.2 Test Cases (input data and expected results)	
<b>4.2 Structural Testing . . . . .</b>	<b>35</b>
4.2.1 Coverage Criteria	
4.2.2 Test data design technique(s)	
4.2.3 Test Cases (input data and expected results)	
<b>4.3 Integration Testing . . . . .</b>	<b>36</b>
4.3.1 Integration strategies	
4.3.2 Test drivers and stubs	
4.3.3 Test Cases	
<b>4.4 Test Evaluation . . . . .</b>	<b>37</b>
4.4.1 Test cases and actual test results	
4.4.2 Structural coverage measurement	
4.4.3 Errors detected and corrected	
4.4.4 Summary of testing experience	
<b>Appendices . . . . .</b>	<b>39</b>
Activity Log. . . . .	<b>39</b>

# 1.0 Introduction

## 1.0.1 Project Statement, Objections, and Functions

Nonprofit organizations have different conditions based on twelve principles. These principles are Advocacy and Civic Engagement; Board Governance; Equity, Diversity, and Inclusion; Financial Management; Fundraising; Human Resources, Information and Technology; Legal Compliance and Transparency; Partnerships and Collaboration; Program Design, Management, and Evaluation; Strategic Communication; and Strategic Planning. Each principles' current condition can be evaluated into three categories: Build, Thrive, or Sustain, where Build is the least mature, Thrive is a middle ground, and Sustain is the most mature. The North Carolina Center for Nonprofits would like for there to be a way for nonprofits to self-evaluate their condition and provide detailed feedback, assessment, and services.

Based on requirements and stipulations set forth by the North Carolina Center for Nonprofits, a survey will be created on SurveyMonkey for nonprofit organizations to fill out. After self-assessing their progress on each of the guiding principles, the results will be uploaded to a website application. An advisor will be able view the results and provide any specific feedback before converting the results into PDF format.

This project will be an iteration of the previously created Basic Assessment tool. This version will streamline the process, removing the need for an advisor to manipulate Excel files while allowing them to view a client's results and provide feedback in a more intuitive interface as a part of the website application. The PDF generation will be handled automatically in order to make the process as simple and efficient as possible.

# 1.1 Project Estimates

## 1.1.1 Historical data for estimation (previous group)

Estimated completion: 14 weeks

Determine functional requirements	35 hours
Research different survey tools	25 hours
Research integration of survey tool with OneDrive	10 hours
Create prototypes of surveys	10 hours
Integrate survey tool with OneDrive	10 hours
Create final survey	5 hours
Write documentation for users	10 hours

## 1.1.2 Estimation Techniques

COCOMO II (revised version of the constructive cost model)

The COCOMO II is a model that estimates the cost, effort, and schedule of a software project. This model is made up of three sub-models which include end user programming, intermediate sector, and infrastructure sector. The COCOMO II model also consists of three stages. Stage one supports the project estimation of a prototype. To do this it uses the Application Composition Estimation Model. The second stage supports the estimation of the early design and to do so it relies on the Early Design Estimation Model. The last stage supports the estimation of the post architecture of a project. To accomplish this the model uses the Post Architecture Estimation Model.

## 1.1.3 Estimations

### Cost

Estimated Completion: 14 weeks

Review historical data from previous group	35 hours
Communication with NCCNP	15 hours

Research possible tools for implementation	10 hours
Create prototypes	20 hours
Create and update project deliverables	10 hours
Prepare and create final presentation	5 hours

### **Schedule**

Week 1: Form groups, decide group project focus

Week 2: Acquire previous groups work, contact NCCNP, assign team member roles, begin first project deliverable

Week 3: Complete first project deliverable, draft project questions for NCCNP and Dr. Vetter, continue contact with NCCNP, receive project deliverable feedback, make revisions to first project deliverable

Week 4: Set up face-to-face meeting with NCCNP, draft software requirements, assign roles for second project deliverable, begin second project deliverable, make data flow diagram, draft restrictions and limitations

Week 5: Update first project deliverable, complete second project deliverable, receive feedback from second project deliverable, continue contact with NCCNP

Week 6: Update and revise both first and second project deliverable, begin drafting third project deliverable

Week 7: complete third project deliverable, complete updating second and first project deliverable, receive feedback on third, second, and first deliverable, update deliverables, continue contact with NCCNP and Dr. Vetter

Week 8: Begin implementation, begin drafting fourth project deliverable, update previous project deliverables

Week 9: Specify test cases for functional implementation, continue implementation, continue drafting fourth project deliverable

Week 10: Specify test cases for structural implementation, continue implementation, continue drafting fourth project deliverable

Week 11: Develop strategies for integration, begin testing drivers and stubs, continue drafting fourth project deliverable

Week 12: Complete fourth project deliverable, receive feedback for fourth project deliverable

Week 14: Complete final project deliverable, update all previous deliverables, begin preparation for final presentation

## 1.2 Project Risks

Risk	Likelihood	Severity	Risk Mitigation Plan
Scheduling issues among team members	High	Low	Communicate with absent group members after meetings to assign work and responsibilities as needed.
Group member drops the class	Low	Medium	Redistribute the tasks of the dropped member to the rest of the group.
Group falls behind schedule	Medium	Medium-High	Work overtime to get back on schedule.
Requested requirements from the North Carolina Center for Nonprofits change	High	Medium	Make changes as needed and update the existing project
Bugs are discovered in the previous group's existing project	Low	Medium	Take time as a group or assign someone to debug the program.
Bugs occur in the updated version of the project	Medium	Medium	Work to fix bugs as they occur and test the program in incremental stages to avoid having bugs going unnoticed for too long.
Lack of communication from the North Carolina Center for Nonprofits	Medium	Medium-High	Continue to attempt to get in contact with a representative while continuing to work on the requirements they have already given.
Loss of documentation or code	Low	High	Make sure code, data, and documentation is saved in multiple locations to avoid any setbacks.

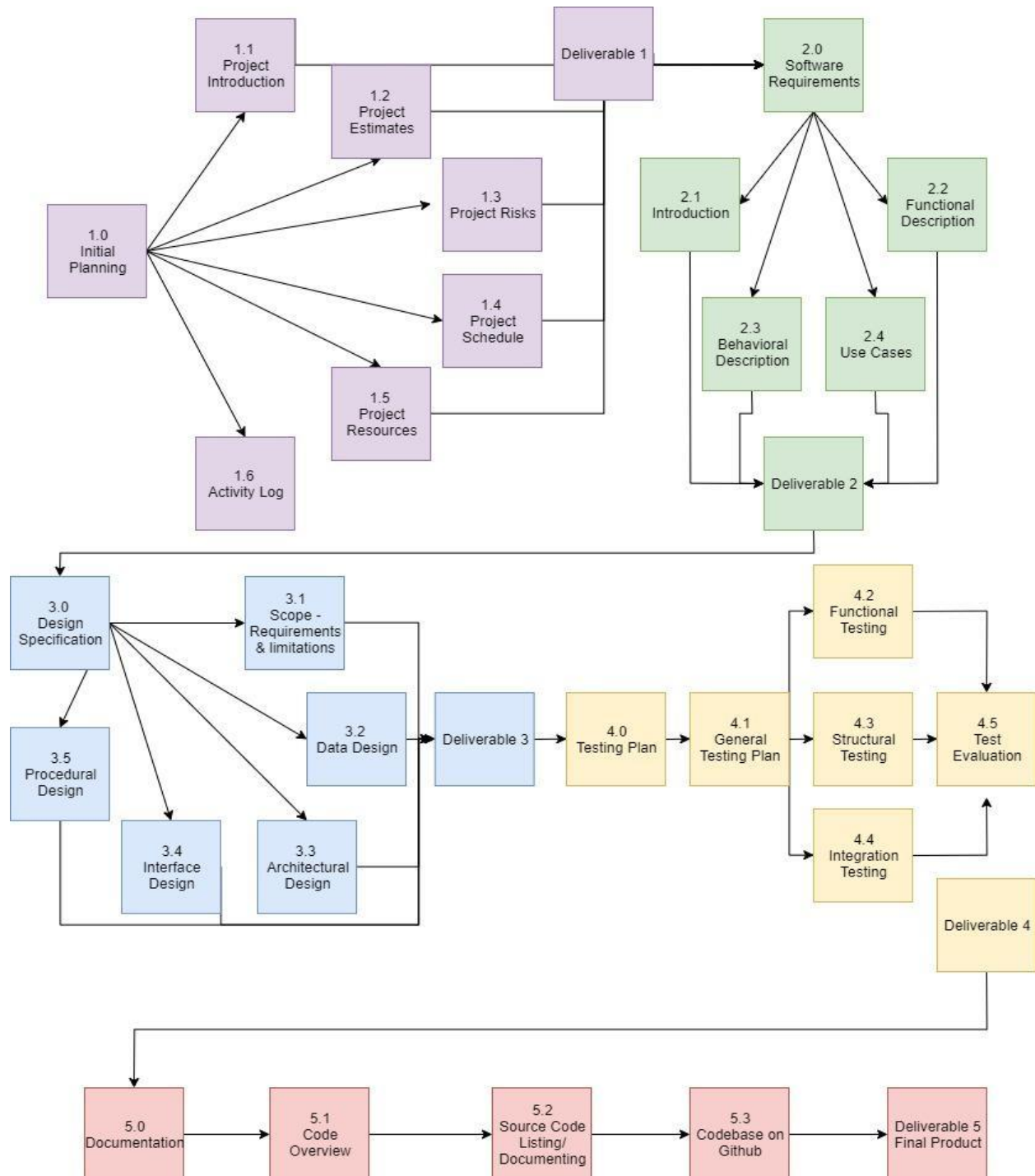


## 1.3 Schedule

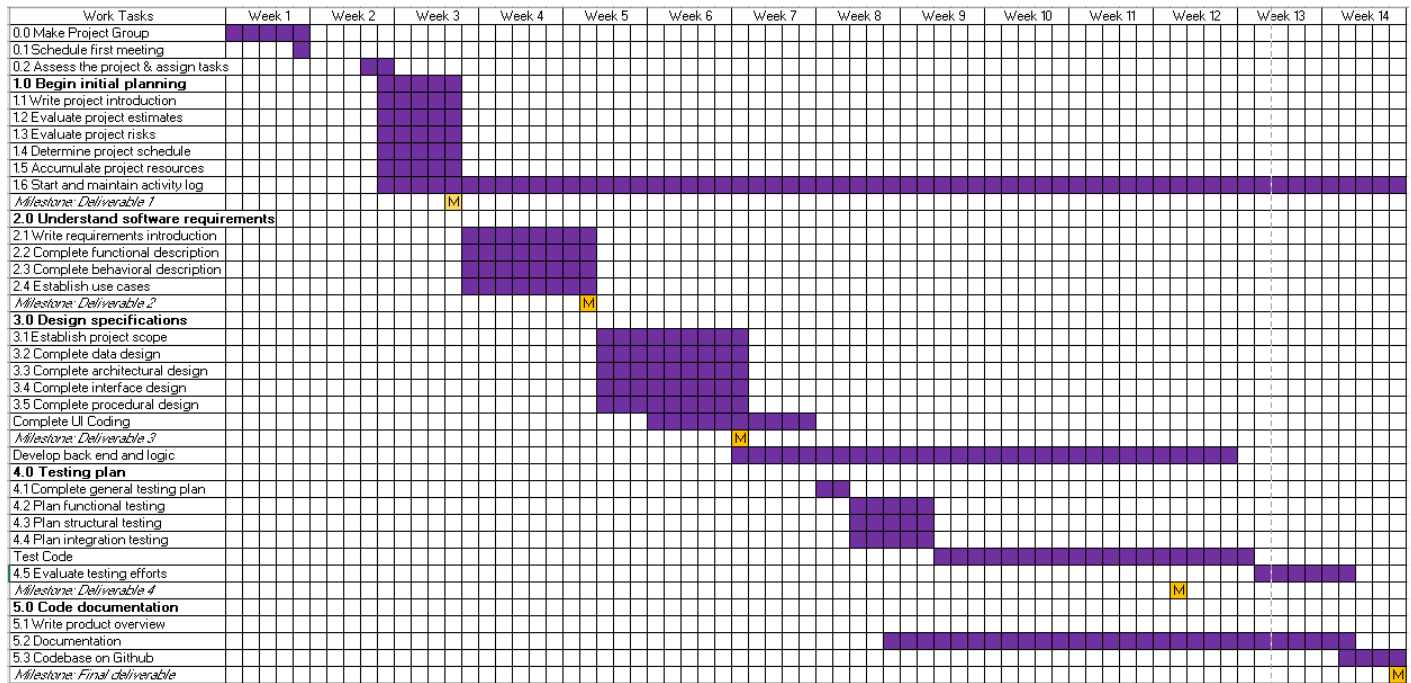
### 1.3.1 Project Work Breakdown Structure

Week	Progress
1	Group formation, project selection, meeting planning
2	Independent overview of former group's work
3	Complete deliverable 1 and contact NCCNP for planning and approval
4	Finalize the final product and decide on a coding language
5	Complete deliverable 2, finalizing software planning
6	Start planning deliverable 3 and begin bare bones coding and organization
7	Complete deliverable 3 and update deliverables 1 and 2, finish coding UI layout
8	Begin work on back end/functionality and authentication (if needed)
9	Establish functional test case specifications, finish coding and tidy it up
10	Establish structural test case specifications, begin testing
11	Finalize integration strategies
12	Complete deliverable 4, testing evaluation
13	Document source code and make sure all deliverables are up to date
14	Complete deliverable 5, final deliverable, and product demo

## 1.3.2 Task Network (CPM)



### 1.3.3 Time-Line Chart (Gantt Chart)



This Gantt chart details our time management planning for the duration of this project. This planning mechanism will help us know when to work on deliverables and when to work on the software itself.

# 1.4 Project Resources

## 1.4.1 People

### North Carolina Center for Nonprofits (NCCNP)

- Salima Thomas
- Rebekah Beck
- Katrina Pareja
- Pamela Palmer

### University of North Carolina Wilmington

- Dr. Ron Vetter
- Natasha Davis

### Team Members

- Audrey Bunn
- Kyle Dove
- Justin Gill
- Eve Neal
- Kevin Todd
- Steven Tucker

## 1.4.2 Hardware

- Mobile phones: Downloaded Slack app for faster communication within team members.
- Personal laptops: Most of the work was completed on personal laptops.
- Whiteboard: Physical space used to lay out ideas when the team holds meetings.

## 1.4.3 Software

- Slack: Slack is used for scheduling meetings and communicating within the team members.
- Trello: Trello is used for managing project progress, splitting up assignments, keeping a to-do list, and assigning individual roles.
- Microsoft Word Online: Word Online is used to collaborate on project deliverables within the team members.
- Email: Email is used to communicate with all parties involved on the project and direct communication with NCCNP.

- Survey Monkey: Survey Monkey is used to get survey results and correspondence with individual non-profit organizations.
- Lucid Chart: Lucid Chart is a website used to create the data flow diagram for the data used in Functional Description (Section 2.2).
- Draw.io: Draw.io is a website used to create the use case diagram for the data used in the Use Case Diagrams (Section 2.4.3)

## 2.0 System Analysis

### 2.0.1 System Requirements

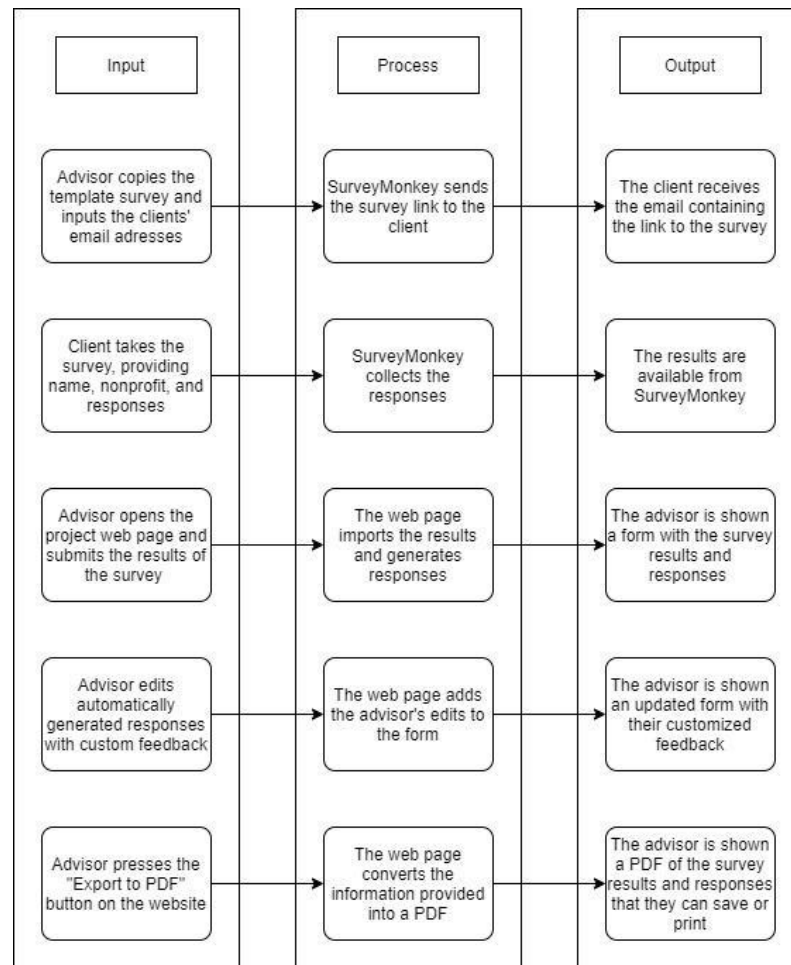
The previous project used a survey that was hosted by Survey Monkey and with additional files stored on SharePoint. A computer with internet access would pull the results of the survey as a Microsoft Excel File. The program that converts the Excel file into the final PDF requires a computer with a Windows Operating System to run and a PDF viewer is required to view the final report.

The proposed update to the project will require a computer with a web browser to interface with the program and the requirement to view and edit an Excel file will no longer exist.

### 2.0.2 Software Project Constraints

The time limit to complete the project is one semester. The project must produce reports that fall in line with the client's expectation and standards, particularly with their official style guidelines. The project must build off the existing work done by the previous group. The project must not incur any costs that are unacceptable for the North Carolina Center for Nonprofits. The current costs incurred are the costs of using Survey Monkey and hosting the SharePoint server. At this point, the NCCNP is unwilling to add any additional monetary costs to the project.

## 2.0.3 System Diagram



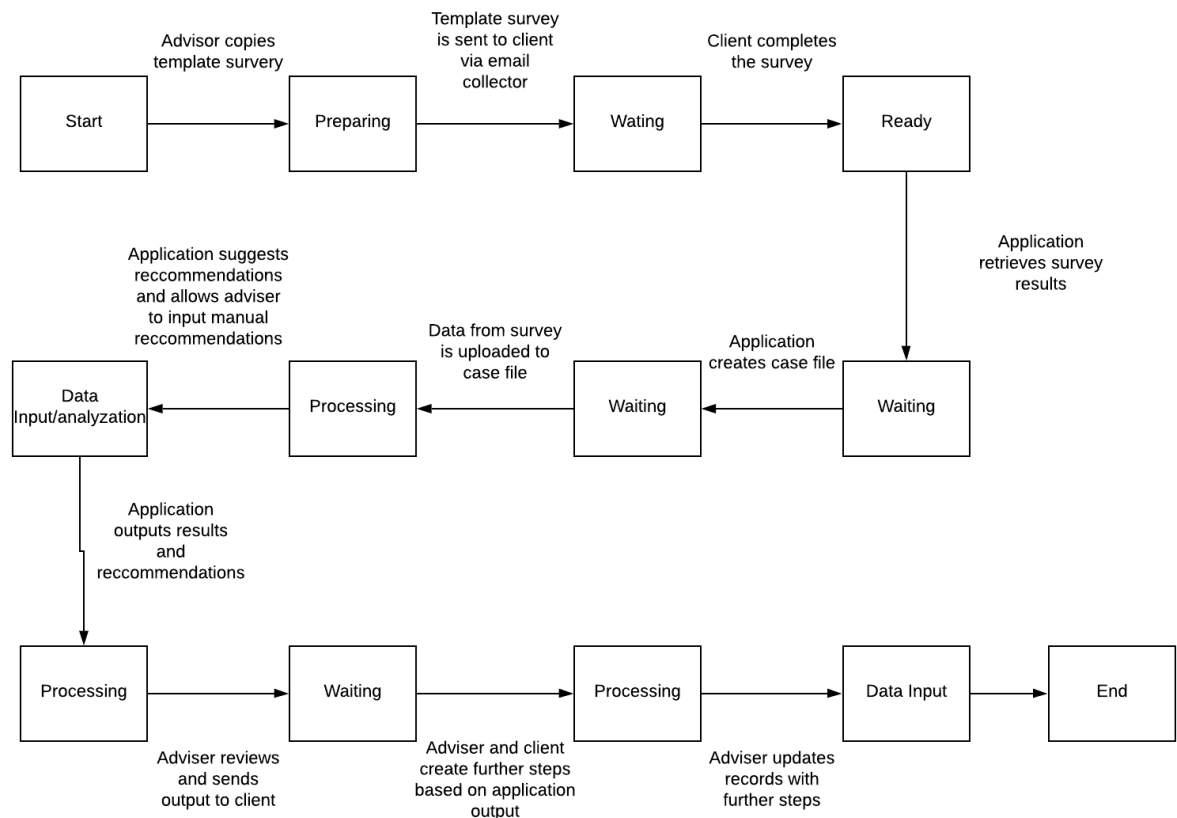
## 2.1 Behavioral Description

### 2.1.1 System States

As seen in the State Transition diagram, different states will be reached depending on what action needs to be met. Some of those states include preparing, waiting, ready, data input, and processing. The actions that take the flow to different states are more specific for this project. The states in the diagram below can be identified by being within a box.

### 2.1.2 Events and Actions

Events and actions are the project specific steps that take the process from one state to another. Some events and actions from this project include the client taking the survey, the application creating a case file, or the application generating suggestions for the adviser and client. Events and actions within the state transition diagram below can be identified by the text surrounding the arrows.





## 2.2 Functional Description

### 2.2.1 Functional Partitioning

The entities are the NCCNP and individual nonprofit representatives. The database in use is SurveyMonkey. The processes are the representatives taking the self-evaluation surveys and formatting the data from SurveyMonkey.

NCCNP creates self-evaluation surveys and pushes them to SurveyMonkey. Individual nonprofit representatives log in to SurveyMonkey and take the surveys. Their answers are recorded to SurveyMonkey. The answers are exported into the formatting tool and processed into a PDF file for easy viewing. The PDF is sent to NCCNP, who then makes it available for viewing on their website. The PDF may be viewed in the browser window or downloaded the browser window or downloaded.

### 2.2.2 Functional Description

NCCNP has internal questions that are compiled into a survey. The survey is for nonprofits to gauge what stage they are in. NCCNP has defined these stages as “Build”, “Thrive”, and “Sustain” and nonprofits progress linearly through these stages as the organization matures. The survey is hosted on SurveyMonkey, a website where surveys are streamlined to interact with the survey creators and takers.

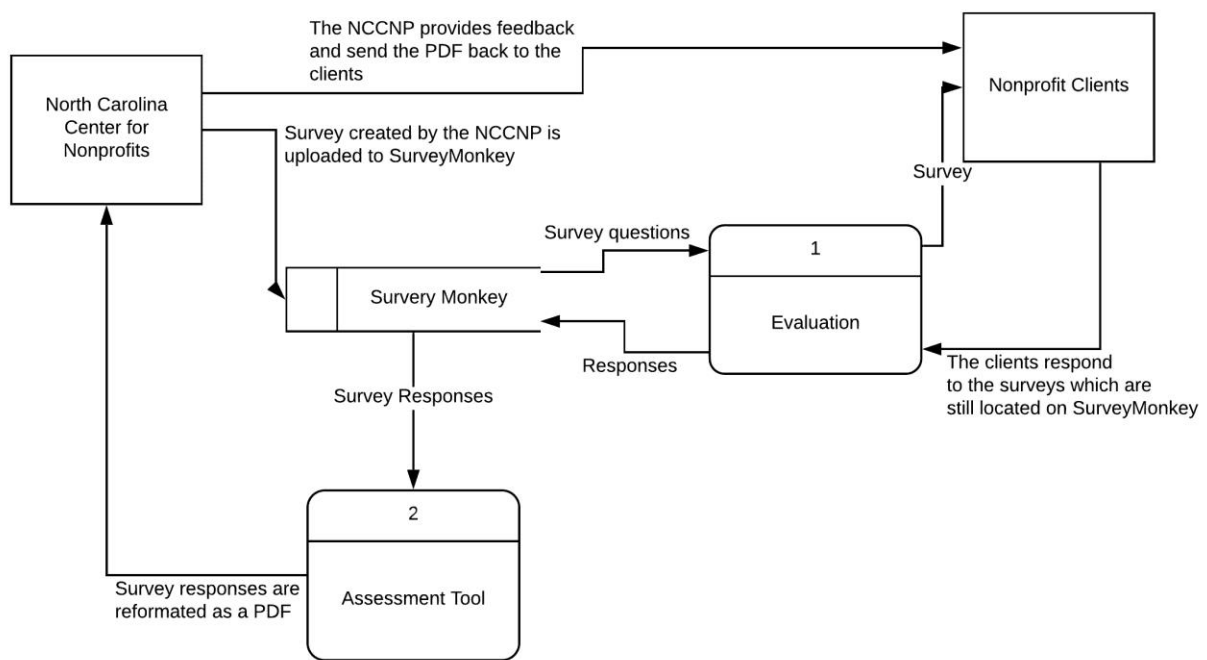
The nonprofit representative logs into SurveyMonkey and selects the NCCNP survey. The representative completes the survey and the results are stored in the SurveyMonkey database. The representative logs into NCCNP and navigates to the survey result page that we are creating. Upon visiting the page, a request is sent to SurveyMonkey to fetch the results of the survey. These results are passed to the formatting tool, which formats the data into a presentable PDF sorted by question category. This PDF is returned to the NCCNP site, where it is presented on-screen. There will be a button to save the PDF as a file to the representatives' computer or it can be revisited in the browser.

Since we do not have direct access to the server that they run their website on, NCCNP would not be able to see the product on their end until we hand over the files for our web design. We also do not know how or where they store information about different nonprofits since that information is all internal.

Depending on how the SurveyMonkey account is set up, there may be API request limitations affecting how many surveys can be pulled per minute. To get around this, the limit may be increased for a fee. Alternatively, training for the NCCNP employee reviewing the survey will have to include the request limit. The employee must keep this limit in mind when pulling several surveys successively. Another limit is the 15-week time period of the semester. This project must be fully developed to encompass the needs of NCCNP to the

best of our ability by the end of the semester. With the project being part of the NCCNP website, the formatting of the software will need to be consistent with their design standards. This includes specific color hex values for the logo, font specifications, and wording choice.

The primary users of this software are looking for something simple and not confusing. The webpage should have a straightforward layout with as few buttons as possible to minimize steps needed to view results. Performance is not a huge concern, but we would not want to take more than several seconds to load a webpage. We need to do thorough error handling to make sure the reviewer is alerted if a survey cannot be found or other basic errors that may arise.



### 2.2.3 Control Description

Design constraints around the project include the fact that the assessment tool must integrate SurveyMonkey into the design and all the data that will be formatted is coming from there. The data must be reformatted into a PDF that meets the desired standards of the North Carolina Center for Nonprofits. Another constraint is that design of the application depends on what the North Carolina Center for Nonprofits will allow or is capable of implementing, in that it may be an application hosted on their website if the company that hosts their site allows it, otherwise it will need to be an application that is downloaded.

## 2.3 Use Cases

### 2.3.1 Stakeholders

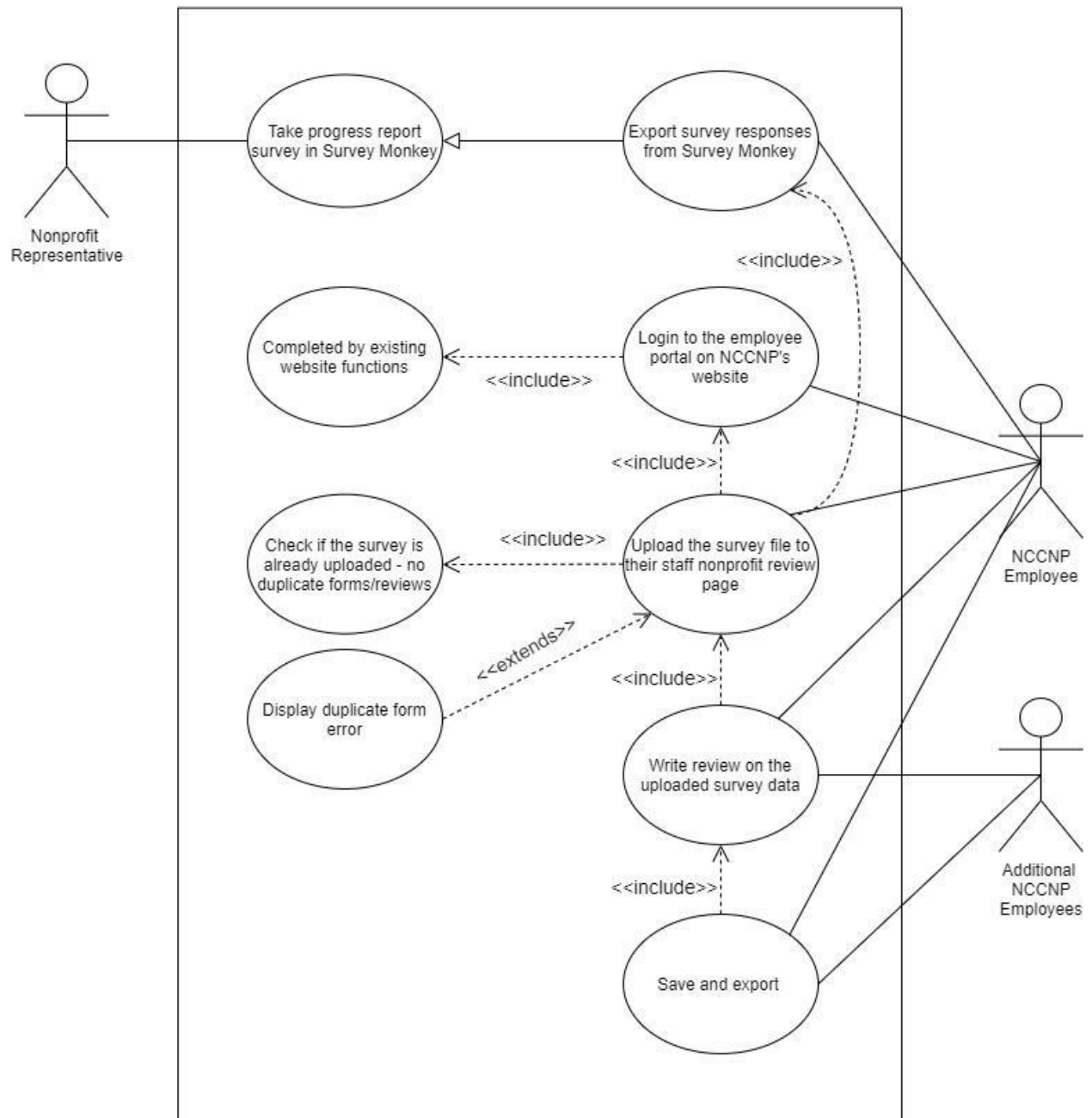
In this case, both the end users, Nonprofit Representative, and providers, the NCCNP Employee and any other NCCNP Employees, are stakeholders as they both will be affected by the outcome of the product. The Nonprofit Representative will use the program to take a progress report survey in which the NCCNP Employee will export those responses and review the information given while other NCCNP Employees will write a review on the survey data.

### 2.3.2 Actors

The main actor would be the NCCNP Employee followed by the Nonprofit Representative NCCNP Employee due to how much interaction and input they give to the system. The Nonprofit Representative will give input based on the survey. Followed by the review and writing of information based on said survey.

### 2.3.3 Use Case Diagram

#### NCCNP Nonprofit Recommendation Application



### 2.3.4 Use Case Description

Use Case	NCCNP employees review nonprofits' status reports																	
Goal in Context	A new status report is submitted on Survey Monkey and NCCNP must review the data and return recommendations.																	
Scope & Level	NCCNP, Primary Task																	
Preconditions	The nonprofit has submitted a Survey Monkey status report																	
Success End Condition	NCCNP completes reviews and recommendations and exports data as a pdf																	
Failed End Condition	NCCNP cannot complete and or export reviews/recommendations																	
Primary & Secondary Actors	Primary – Nonprofit agency submits form Secondary – NCCNP employees review form data and write recommendations																	
Trigger	Survey Monkey survey is submitted																	
DESCRIPTION	<table><tr><th>Step</th><th>Action</th></tr><tr><td>1</td><td>Nonprofit submits form</td></tr><tr><td>2</td><td>NCCNP exports survey data</td></tr><tr><td>3</td><td>NCCNP logs into employee portal</td></tr><tr><td>4</td><td>NCCNP uploads survey data into our reviewing web app within their website</td></tr><tr><td>5</td><td>NCCNP employees write recommendations in the web app and saves the data</td></tr><tr><td>6</td><td>Once enough recommendations are written, an employee exports the page and saves it as a pdf</td></tr><tr><td>7</td><td>NCCNP employee sends the pdf to the nonprofit</td></tr></table>		Step	Action	1	Nonprofit submits form	2	NCCNP exports survey data	3	NCCNP logs into employee portal	4	NCCNP uploads survey data into our reviewing web app within their website	5	NCCNP employees write recommendations in the web app and saves the data	6	Once enough recommendations are written, an employee exports the page and saves it as a pdf	7	NCCNP employee sends the pdf to the nonprofit
Step	Action																	
1	Nonprofit submits form																	
2	NCCNP exports survey data																	
3	NCCNP logs into employee portal																	
4	NCCNP uploads survey data into our reviewing web app within their website																	
5	NCCNP employees write recommendations in the web app and saves the data																	
6	Once enough recommendations are written, an employee exports the page and saves it as a pdf																	
7	NCCNP employee sends the pdf to the nonprofit																	
EXTENSIONS	<table><tr><th>Step</th><th>Branching Action</th></tr><tr><td>4a</td><td>The survey responses have already been uploaded by another employee 4a1. Display error message</td></tr></table>		Step	Branching Action	4a	The survey responses have already been uploaded by another employee 4a1. Display error message												
Step	Branching Action																	
4a	The survey responses have already been uploaded by another employee 4a1. Display error message																	

## 3.0 Scope

### 3.0.1 System Objectives

For this application, system objectives should be clear, concise, and realistic. Given the recent COVID-19 outbreak and the nature of school now, our objectives must be appropriate and realistic. Our system objects are to streamline the process of analyzing, sharing, and utilizing surveys to help non-profits reach their goals. This entails expanding upon the work already laid out from the previous group to work on this system. Another major system objective of this project is to do this on a web application which requires us to complete the tasks in HTML, CSS, and PHP with an approachable interface and a system with reliable functionality.

### 3.0.2 Major Software Requirements

Given the objectives and goals we are pursuing, certain software requirements are created. Firstly, the software must have an interface and visual appearance which is in line with the branding standards of the NCCNP. Taking into consideration our users, the software must have an interface which is clear, usable, and approachable. Our software must be founded in the ten-step process of the NCCNP and be orientated towards the desired goals. The software must be coded with languages that are reasonable to create a web application. Lastly, the software must be done in a way that it could be passed on to a future group to expand upon.

### 3.0.3 Design Constraints and Limitations

One of the biggest design constraints and limitations at this point is the number of workable hours. With COVID-19 taking away a week from working and all future face-to-face meetings the process has been slowed down. This is a constraint that must be taken into consideration going forward with the design process. Another constraint is working with the NCCNP. While working with a real-world client is great experience and meaningful, most decisions must be approved and discussed before any design work can be done.

## 3.1 Data Design (Entity Relationship Diagram)

### 3.1.1 Data Objects

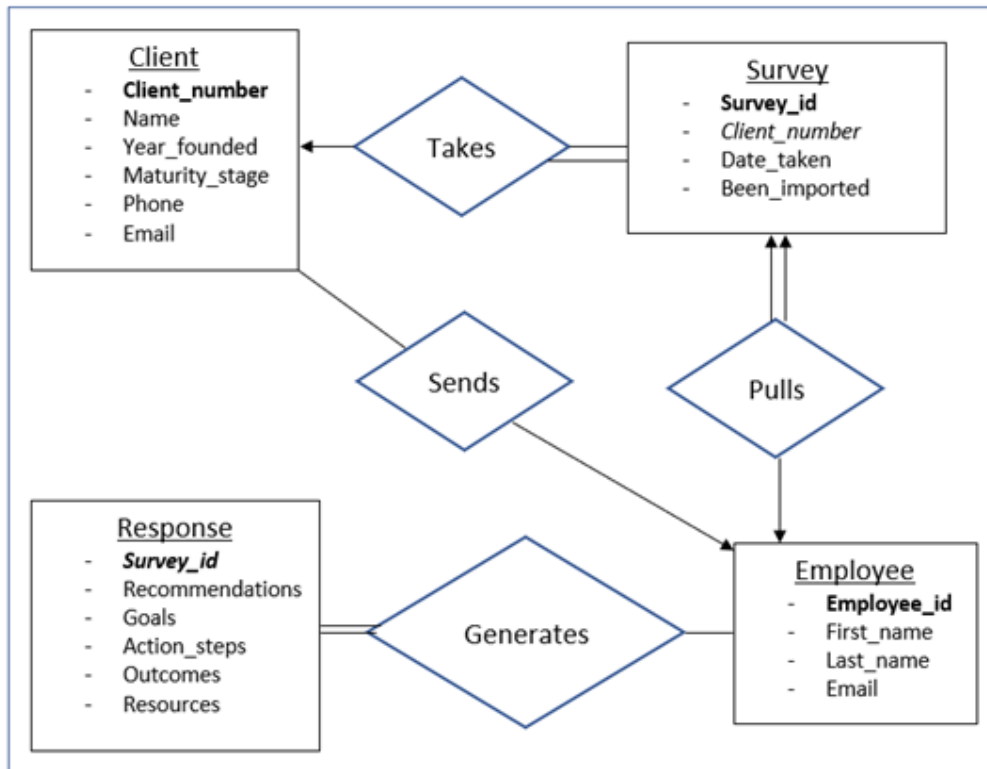
The involved entities are the survey, the nonprofit client, the NCCNP employee, and the responses to the survey. The following are entity sets of information about each entity:

- Client = (client\_number, name, year\_founded, maturity\_stage, phone, email)
- Survey = (survey\_id, *client\_number*, date\_taken, been\_imported)
- Employee = (employee\_id, first\_name, last\_name, email)
- Response = (survey\_id, recommendations, goals, action\_steps, outcomes, resources)

Primary keys are underlined. Foreign keys are italicized. Client\_number in Survey is a foreign key from Client. Survey\_id in Response is a foreign key from Survey.

### 3.1.2 Relationships

The following table is an entity-relationship diagram that shows the entities and relationships. Each entity has their associated attributes outlined and each relationship shows the participation and cardinality. A more in-depth description of each relationship follows the diagram.



The relationship Takes(survey, client) is a many-to-one relationship. Many surveys may be taken by one client. Survey has total participation and the client has partial participation. Each survey must be taken by a client, but not all clients must take a survey.

The relationship Pulls(survey, employee) is a one-to-one relationship. One employee may pull one survey at a time and the survey may be pulled only once. Survey has total participation and the employee has partial participation. Each survey must be pulled by an employee, but not all employees must pull surveys.

The relationship Generates(employee, response) is a many-to-many relationship. Many employees can generate one response and one employee can generate many responses. Any number of employees may contribute to the generation of any number of responses. Employee participation is partial and response participation is total. Each response must be generated by at least one employee, but not every employee has to generate a response.

The relationship Sends(employee, client) is a one-to-many relationship. One employee can send multiple responses to a client and multiple clients can be sent their response by one employee. Employee participation is partial and client participation is partial. Not every employee will send a response. Not every client will receive a response. Only the clients that have taken a survey will receive a response.



## 3.2 Architectural Design

### 3.2.1 Review of Data and Control Flow

We will use the Data Flow architectural design as it will be easier for NCCNP to implement and incorporate into their system. Information will be passed through filters that will change the data as it goes down every stage of the process.


### 3.2.2 Derived Program Structure

1. Survey: The Nonprofit organizations will respond to a survey hosted by SurveyMonkey.
2. SurveyMonkey: SurveyMonkey will host and collect all the responses and export them into a .csv file format.
3. Website: Information will be hosted on NCCNP's website where employees can upload the .csv file from SurveyMonkey.
4. TinyMCE: From there, the employee can create recommendations and save.
5. PDF: There will be a button that allows the employees to export the recommendations as a PDF and send them to the Nonprofit Organizations.

## 3.3 Interface Design

### 3.3.1 Human-machine Interface Specification

North Carolina



Center for Nonprofits

Upload

Nonprofit	Last Edited	
Soup Kitchen of New Hanover County	N/A	Edit
Meals on Wheels	1/19/2020	Edit
Feline Hope	1/15/2020	Edit

#### Upload Screen:

This is where exported Survey Monkey surveys will be uploaded to our application for recommendations to be added. The user will click upload and the file browser on their computer will open so they can upload the exported .csv files from Survey Monkey. Once a survey is uploaded it will appear on the list of editable/recommendable surveys. When the user clicks an edit button it will take them to the edit screen for that survey.

North Carolina  
Center for Nonprofits

Type your name here to save... Save Export

**Advocacy and Civil Engagement**

Current Condition: Build

Desired Condition 1 y from now: To progress from build to thrive

Recommender 1 Name: This nonprofit needs to.....

Recommender 2 Name: In order to progress to a thriving state.....

Type your recommendation here...

**Board Governance**

Current Condition: Build

Desired Condition 1 y from now: To progress from build to thrive

Recommender 1 Name: This nonprofit needs to.....

Recommender 2 Name: In order to progress to a thriving state.....

Type your recommendation here...

### Edit Screen:

The edit screen will have the nonprofit's survey responses on the left hand side and recommendations on the right hand side. The left side will be uneditable and the right side will be partially editable. Existing recommendations from other NCCNP employees will be on the right-hand side and will not be editable. Below the uneditable existing recommendations will be a text box for the user to make their recommendations. The text box will utilize the open source editing tool TinyMCE to incorporate more text editing capabilities like hyperlinks.

The save button will save the text that has been entered and make it uneditable like the existing recommendations. In order to save, the user must type their name in the text box next to the save button so their recommendation will be attached to their name for other NCCNP employees to see. The export button will call window.print and bring up the browser's print menu. From there, the user can select "Save as PDF" from the printer drop down box and save the page as a css formatted PDF within company styling guidelines.

### 3.3.2 Human-machine Interface Design Rules

1. Keep the page simple, clean and easy to use with as minimal steps as possible.
2. The nonprofit survey info and recommendations on the edit screen must be clearly divided on two halves of the screen for usability.

3. The editable text area on the edit screen must be clear. The user should be immediately aware of where they can type on the screen.
4. All styling must be cohesive with NCCNP's current branding.
  - a. Font must be Lato
  - b. NCCNP brand colors must be present
  - c. The NCCNP name and logo should be on exported pdfs
  - d. Pdfs should include Respondents' and their Advisor's name and contact information

### 3.3.3 External Interface Design

#### External Systems

- Nonprofits will complete progress report surveys on Survey Monkey.
- If NCCNP does not want our program hosted on their existing webpage and we create an offline web page it will need to be stored on a cloud storage service like OneDrive. That way, all saved data will be appended to the same shared text file and all user's data will be synced.

#### Data Storage:

- **Survey Data** will be stored in a "surveys" file within the web app file structure. When the upload page loads, it will pull file names from the surveys folder to populate the list of uploaded surveys. When "edit" is clicked and the edit page loads it will pull the data from that specific svg file in the surveys folder.
- **Recommendation Data** will be stored in text files since we do not have access to a database or server. Each .csv file in the surveys folder will have a corresponding text file where recommendations will be stored. Recommendations will be entered into text files in an html text format so they can be displayed without processing once the edit page loads. When a new recommendation is saved, it will be appended to the end of the text file.
  - Ex.) John Doe: This nonprofit should spend more of their budget on marketing. \n Jane Doe: Publicity will help this nonprofit's fundraising efforts. \n

### 3.3.4 Internal Interface Design

#### App File Structure:

- **styles**
  - Upload.css
  - Edit.css
- **surveys**
  - foodpantry.csv
  - cancerresearch.csv
  - habitatforhumanity.csv
- **recommendations**
  - foodpantry.txt
  - cancerresearch.txt
  - habitatforhumanity.txt
- Login.html
- Upload.html
- Edit.html

#### Component Intercommunication:

- **Upload.html** retrieves filenames within the “surveys” folder and populates the list of uploaded surveys. It also saves survey files that are uploaded via the “upload” button to the “surveys” folder and repopulates the survey list. When a new survey file is uploaded, a new text file is created in the “recommendations” folder to store future recommendation text.
  - surveys – read existing filenames, add new file
  - recommendations – add new
  - styles – link to corresponding .css file
- **Edit.html** populates the left side of the screen (nonprofit survey data side) with the data in the selected nonprofit’s survey .csv file from the “surveys” folder. It populates the right side of the screen (existing recommendations side) with the text data in the current survey’s .txt data file from the “recommendations” folder. When the user types their recommendation in the text box and clicks save, the Edit.html page saves the user’s text input in the same .txt file that the page pulled existing recommendation data from.
  - surveys – load file data
  - recommendations – read/write to existing file
  - styles – link to corresponding .css file (the same .css file has the output pdf formatting)

## 3.4 Procedural Design

### 3.4.1 Processing Narrative

Upon starting up the Nonprofit Condition Assessment tool the user will be prompted to login into the tool before being able to edit any assessments. Once the user is logged in, they will be brought to the upload screen where the user will have the option to either upload .csv files that were exported from SurveyMonkey or edit files that were already uploaded. Selecting to edit files will take the user to the edit screen. The left-hand side of the screen cannot be edited and contains survey responses. The right-hand side is where the user will be able to fill out their recommendations and see existing recommendations. The user will also have the option to save the edits they make or export the edited file as a PDF.

### 3.4.2 Interface Description

The first page of the interface uses upload.html to allow the user to upload the .csv for the surveys taken from SurveyMonkey to be edited. The user will have the option to edit the surveys which takes them to the second interface page which uses edit.html. This allows the user to edit the survey by filling out recommendations or exporting the edited survey as a PDF.

### 3.4.3 Design Language Description

If upload is selected

- If file uploaded is a .csv

  - Survey uploaded to be edited

- Else upload a different a .csv file

If edit is selected

- Survey may be edited with recommendations

- If save is selected

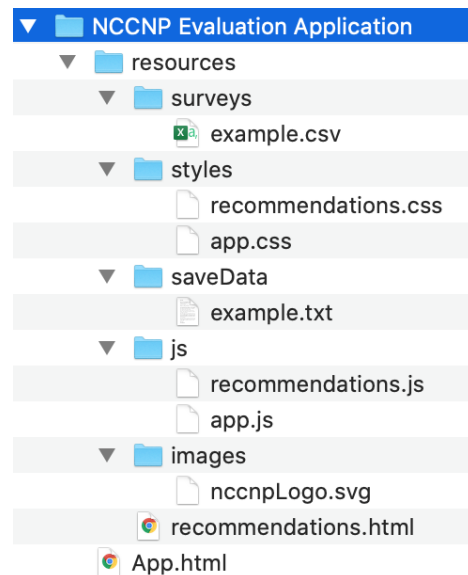
  - Edits to the survey are saved

- If export is selected

  - Survey is reformatted and saved as a PDF

### 3.4.4 Modules Used

Our file structure for this project is not contemporary for the purpose of abstracting the behind the scenes files and data to make a less complex user interface. Since our html, css, and javascript files will be saved to the NCCNP OneDrive, we need to hide our visible resource modules from the user. This is our file structure:



Our application is split into two modules, one for each page. The first page lets the user upload Survey Monkey files, view their list of uploaded surveys, and select surveys to edit and add recommendations. This module is comprised of the outer App.html page that launches the application for the user; as well as an app.css file for styling and an app.js file to handle logic. When a user uploads a .csv file from Survey Monkey on page one the file is saved within our file structure to the “surveys” folder. When app.html is loading, the list of uploaded surveys is populated from the filenames in the “surveys” folder.

The second module is for the second page and it has a similar set up to the first page. When a user clicks edit on page one, they are taken to recommendations.html, which is where recommendations can be added and edited. The second page’s html page is hidden within the resources folder because a user must start the application from page one. Page two is created with recommendations.html, recommendations.css, and recommendations.js. When a user finishes writing their recommendations the text they entered is encoded and saved to a corresponding text file in the “saveData” folder. The recommendations.css file is responsible for formatting the final product, the

recommendation pdf file that NCCNP will distribute back to the Nonprofit that filled out the survey.

The two modules, page one and page two and their respective helper files are relatively separate from each other, other than the survey data on page two. When page two is loading, it pulls survey responses from the “surveys” folder to populate the left side of the screen with Nonprofit responses to the Survey Monkey survey.

### 3.4.5 Internal Data Structures

Javascript arrays and associative arrays for storage of information from text files.

### 3.4.6 Comments/Restrictions/Limitations

The major limitation to this entire program is that it will only work with .csv files, but since the North Carolina Center for Nonprofits only uses SurveyMonkey this will not pose any issues for them. The files for the program must also be downloaded since it is not being hosted on the NCCFNP website.



## 4.0 Test Plan

### 4.0.1 What to Test

When testing the web applications certain areas needed to be focused on and tested more than others. Those areas included functionality, usability, and appearance. Functionality needed to be tested fully because the client needed the software to be able to complete tasks that were specific to the nature of their business. With the future users not being computer science students or majors, the usability needed to be simplistic and easy to navigate. This required the testing of usability to be thorough as well. Lastly, with customers of the client also engaging with this software, the importance of appearance and branding was a priority for the client which meant the testing of the appearance was crucial to the reception of the software.

### 4.0.2 When to Test

Since most testing was dependent on the feedback of prototypes, the time for testing came right before and right after feedback. Before each prototype was presented, testing had to occur to ensure that all aspects regarding functionality, usability, and appearance were addressed and met. After feedback was given on the current prototype, more testing would occur to see where the necessary change could be made to meet the feedback.

### 4.0.3 How to Test

The testing of the functionality of the software could be done easily without writing extra test scripts. After different functionalities were added into the software the testing would consist of going into the application on the user's end and entering in data as if it were from a user. All usability and appearance testing were done by providing a prototype to the client and receiving feedback on the usability and appearance and then updating appropriately.

## 4.1 Functional Testing

### 4.1.1 Test Data Design Technique(s)

Since functionality was the most important aspect that needed to be tested the black-box testing method was used. The representatives from the North Carolina Center for Nonprofits demoed the program and tested its functionality without ever looking into the internal structure of the application.

### 4.1.2 Test Cases (input data and expected results)

The data that was used were example survey files from SurveyMonkey. They were put through the application to test to see if the formatting was done correctly and in a way that met the needs of the North Carolina Center for Nonprofits. The expected results were a correctly formatted PDF.

## 4.2 Structural Testing

### 4.2.1 Coverage Criteria

To account for structural testing our testing coverage was done alongside and complimentary to the functional testing. Along with feedback from the North Carolina Center for Nonprofits on the web application, the developers of the program went through and input dummy data into the web application to see how it was treated by the source code. The structural testing went through each page of the web application and the backend as well. This coverage allowed the developers to find hidden errors in the code outside of syntax errors and find the dead code within the source code that was not contributing to the web-application.

### 4.2.2 Test data design technique(s)

The main technique used for structural testing was the developers inputting dummy data into the web application to observe and analyze how the source code would react. For example, the developers would create a fake survey on survey monkey and input survey monkey's output into the web application as a survey. Then observations on the home page would be made to see if the source code reacted as inspected. The same technique was used and carried out for all other pages to test the structural aspect of the web application.

### 4.2.3 Test Cases (input data and expected results)

The inputted data for all structural testing initially came from Survey Monkey because that is the same type of input that will come from the NCCNP. From there all input was given by the user which in this case was the developer inputting fake example data that would be representative of what the actual user would be inputting. The expected results for these test cases were that the pages were updated correctly (regarding structure), that the necessary functions and structures would become available to the user, and that the source code structure was handling the input correctly and as expected.

## 4.3 Integration Testing

### 4.3.1 Integration strategies

The big bang integration testing strategy was used since all the components of the program were tested all at once in one go by the North Carolina Center for Nonprofits.

### 4.3.2 Test drivers and stubs

Since the integration strategy was the big bang method there was not a need for drivers or stubs since each unit of the program was tested simultaneously instead of in a bottom-up or top-down method.

### 4.3.3 Test Cases

Objective	Expected Result
Test that the survey and responses are properly formatted.	A PDF to be generated in the proper format of the survey and the responses.
Check that the user can go back in and update or add additional recommendations.	The interface to correctly display any comments that are added or updated.
Check that surveys are properly being uploaded and displayed properly.	Surveys will appear in the list of uploaded surveys and the responses will be displayed in the correct format when they are opened in the tool.

## 4.4 Test Evaluation

### 4.4.1 Test cases and actual test results

Test cases include situations that we expect the software to be under in common situations.

Test	Goal	Result
Input Survey	System successfully adds survey and all functions such as the ability for the user to add comments to each of the pages.	Successful
Add comments to a page	Comments are saved and visible to any users alongside the user that added them.	Successful
Other users' inputs into the system	Other users can add their own comments and still see comments from other users.	Successful
Export to PDF	Successfully save the responses as a PDF	Successful
Searching the list of nonprofits	Search the list of nonprofits to find the nonprofit in need to be reviewed	Successful
Saving without typing your name	Saving your comments without typing in a name will cause an error	Successful
Going from page to page with the navigation links	Ensure the links work and the program will still perform and not reach an unexpected error.	Successful
Collapsible columns in goals and action step.	Ensure the columns can be collapsed so the user can see the survey responses without a lot of information in between the current comment section and the questions.	Successful

### 4.4.2 Structural coverage measurement

Roughly 80% of the code was covered in the usage of the User Interface testing. Unit testing was not conducted so exact numbers are not available.

### 4.4.3 Errors detected and corrected

- Simple syntax errors such as font incorrect in certain areas.
- Hyperlinks not usable in comments or pdf format.
- PHP form submitted comments twice if the page was refreshed.
- Columns were not collapsing on the Goals and Action Steps

All listed errors were corrected.

#### 4.4.4 Summary of testing experience

Due to the constant testing for every use case, it was frustrating trying to find the solution to an error and changing the code. But once finding the solution, it was gratifying to get everything working without compromises.

# Appendices

## Activity Log

### Meeting 1/25/2020

#### Meeting format

Group members met face to face in Randall Library in room 1045

#### Attendants

- Audrey Bunn
- Kyle Dove
- Justin Gill
- Eve Neal
- Steven Tucker

#### Notes

- Organize responsibilities for our first Deliverable
- Choose a member, Justin Gill, to be the direct Liaison with NCCNP
- Get Trello set up
- Converse about what the project is and where the previous group left off

### Meeting 1/28/2020

#### Meeting format

After class

#### Attendants

- Audrey Bunn
- Kyle Dove
- Justin Gill
- Eve Neal
- Kevin Todd
- Steven Tucker
- Dr. Vetter

#### Notes

- Collect all components and combine the data
- Rough idea of what our next steps are

## Meeting 2/5/2020

### Meeting format

Video Call with NCCNP

### Attendants

- Audrey Bunn
- Kyle Dove
- Eve Neal
- Kevin Todd
- Steven Tucker
- Salima Thomas
- Rebekah Beck
- Katrina Pareja

### Notes

- Introduction to NCCNP
- Goal of working together
- PowerPoint of our thoughts and what their team thought of it was that there is a possibility of this working
- Shared understanding of the format and layout of the final output.
- Scheduled biweekly meetings with NCCNP on Wednesdays at 12:30pm

## Meeting 2/13/2020

### Meeting format

After Class

### Attendants

- Audrey Bunn
- Kyle Dove
- Justin Gill
- Eve Neal
- Kevin Todd
- Steven Tucker
- Dr. Vetter

### Notes

- Assigned duties for Deliverable 2

## Meeting 4/08/2020



## **Meeting format**

Zoom

## **Attendants**

- Audrey Bunn
- Kyle Dove
- Eve Neal
- Kevin Todd
- Steven Tucker
- Dr. Vetter
- Rebekah Beck
- Dr. Pamela Palmer
- Katrina Pareja
- Salima Thomas

## **Notes**

- Presented current project demo and received valuable feedback.

## **Meeting 4/22/2020**

### **Meeting format**

Video call through Zoom

### **Attendants**

- Audrey Bunn
- Kyle Dove
- Eve Neal
- Kevin Todd
- Steven Tucker
- Dr. Vetter
- Rebekah Beck
- Dr. Pamela Palmer
- Jeanne Tedrow
- Salima Thomas

### **Notes**

- Presented an updated project demo and received valuable feedback and approval on the overall design. Move forward with testing and implementation.

## **Meeting 5/4/2020**

## Meeting format

Video call through Zoom

## Attendants

- Audrey Bunn
- Eve Neal
- Kevin Todd
- Kyle Dove
- Steven Tucker
- Dr. Vetter
- Katrina Pareja
- Rebekah Beck
- Dr. Vetter
- Rob Maddrey
- Natasha Davis
- Salima Thomas
- Amber Lurker
- Tracy Careyette
- Lbrinkley
- Anna Gallagher
- Sarah
- Jeanine Minge
- Dr Pamela Palmer
- J O'Loughlin
- Katrina
- NCCNP
- Rebekah Beck
- THoward

## Notes

- Presented the final project to NCCNP.
- Other students from Dr. Vetter's class was welcome in the meeting.
- Other members from UNCW were present along with many members from NCCNP.
- Feedback Received
  - It would be very useful to be able to sort by categories.
  - "This is exactly where we wanted to go" - Salima Thomas.
  - Large amount of approval received from many other members of NCCNP and other members of the University.