Sprint 1 Phase 3 Planning

## Know the end from the beginning

* Sprint ends March, 14th 2020
* This sprint will probably take 10 hours a person to complete.
* We are planning to meet every Tuesday and Thursday.
* Trello has a point system. Every task is worth a number of points. 1 points is worth 1 hour of work. In Theory we should all have similar points that indicate that we spent similar amounts of time on the project.
* For sprints 2 & 3: List 1 item from your retrospective that you said you were going to do better at that you are going to focus on this sprint.
* Brody Larsen is the Scrum Master
  + Johnathan Kunz
  + Brody Larsen
  + Chris Winwood
  + Kosta Sergakis

## Fill out details for each story

* For the start of the First sprint we have no leftovers from the last sprint/phase.

This Sprint we are planning on doing the following:

* Each story that you bring in has a description with:
* The website will have a slideshow carousel on the main page
  + This should be a small (maybe medium) task
  + The slideshow should rotate through 3 or more pictures. Each picture should show for 8 seconds. (the user may click on a button that goes to a picture, that also resets the 8 seconds)
  + The goal we have is to look and act like Netflix’s slideshow carousal. It needs to rotate through the list of pictures with smooth fade transitions.
  + A list/database is required to store the pictures.

Tasks

* + A script is needed to rotate through the pictures
* As a user, be able to see if a submission was accepted
  + This is a small task
  + When a reservation is sending a message on the website should return feedback on whether the server successfully stored the reservation
  + If the website gives a false positive or a false negative, it will be assumed the feedback system doesn’t work

Tasks

* + The webpage needs to display a message
  + The database or another program needs to check or return that the request was successfully accepted and the tool is reserved for the specific customer
* As a user, be able to see a logo on the header and footer of every page
  + This is a small task
  + A picture of the company ‘s logo needs to be added to the webpages

Tasks

* + If the logo is not added to the header and footer of every webpage, the task is not done
  + A .jpeg or .png or .gif of the logo needs to added to the static file of the Spring file system
  + The .html needs to display the picture of the logo
* Format tools.html to look better
  + A small task
  + The look of the tools page is still being decided on
* As a user, be able to go to a contact page.
  + A medium task
  + The contacts page needs a google maps of where the warehouse is. The Contacts page needs to display the company’s contact information. It also needs a form to send emails to the tool company.
  + The webpage needs to clearly describe what the email is form is doing. We can’t expect the user to know what the random textboxes on the page are for.

Tasks

* + Figure out google maps API so our website can display google maps.
  + Display tool company’s contact info on webpage
  + Create forms that make it easy to send an email to the tool company
* As a user, go to a projects page.
  + This is a medium to large task
  + The projects page is a webpage that has pictures and descriptions of people using the tools they got from the tool rental company to remodel a house.
  + The page should do a good job of promoting the tool company as a help service to the community. Not so much an advertisement of the brand of tools

Tasks

* + Pictures of customers/people using tools to remodel a house need to be gathered.
  + Testimonies need to be gathered so they can be displayed on the website
  + The pictures and descriptions need to be organized on a webpage
* Create a user’s manual
  + Large task
  + The user’s manual should explain how the website works. Personal feelings about the project should be left out
  + The user manual should be thorough about how it builds the database, how information is added to it, how info is pulled from it and explain how the website builds itself

Tasks

* + Understand and explain the website
  + Understand and explain the database
* Unit tests
  + Medium task
  + We have to come up with as many small tests that check all the individual parts of our program work
  + Finished unit tests are tests that pass
* Sign Up Page
  + Small task
  + A webpage that has a form that takes input and creates a new customer object/adds a new customer to the database. A user objects needs a username, email and password.
  + The sign up page should make/add a new user to the database. If a user wants there elevated access they need to remember a username and a password that is associated with their username to prove who they are in theory.

Tasks

* + Create a webpage with a form that takes input from the user and create a new user.
* Sign in page
  + Medium task
  + A webpage that has a form that takes user input to give the user elevated access to the website. The webpage needs a username and password input box.
  + The form should log a user in. However, if random inputs log the user in then things aren’t working. There should also be customer level privileges and admin level privileges
  + Logging in should change a variable on the user object to indicate that the user has elevated access.

Tasks

* + That variable should be transferred to the other pages and not reset when the user traverses to other webpages.
  + The user should also be able to log out
  + The user should automatically be signed out after inactivity
* Sign in Server
  + I don’t know what a sign in server is
* Add and delete objects to Customer profiles
  + Small task
  + Be able to add or remove tools to a list that the customer wants to check out. The tools aren’t being checked out yet just being added to a list so they can check out the tools on the list

Tasks

* + Every customer has a list associated with what tools they want to check out.
  + That list should follow the customer around while they navigate the website.
  + Once the tools are checked out they should be removed from the list and added to the checked out list.
* Tool check out
  + Size estimate by the team
    - Small (< 1 day)
    - Medium (1 day)
    - Large (2 days)
    - Any larger than this should be broken down into smaller tasks
  + Description of what is in scope, what’s out of scope
  + Acceptance criteria in the description stating what it will look like when it’s done
  + Tasks that will be accomplished to complete the story
    - These tasks can be created in the GitHub project as notes (but then convert them to issues so that you can assign a person)
    - Assign a size estimate to each task (S, M, L)
* Stories and tasks should not be assigned to a specific person (unless carrying over from a previous sprint)

## Artifacts

* Stories / tasks are created and on the sprint backlog
* Create a spreadsheet graph for burndown by totaling up the size estimates and setting that as your amount of work left to do. (commit and push to /docs/planning/SprintXBurndown.xlsx)
* Sprint planning document (with the top section information to /docs/planning/SprintX.docx)
  + Attach a screenshot of your Sprint Backlog after planning