

**Design and Development of an Occupational Ergonomic Application to support Low Back
Pain Risk Assessment**

by

Bala Kusuma Nanda Kumar Ravilla

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Computer Science

Auburn, Alabama
December 16, 2017

Keywords: User Interface Design, Web Design, Usability, Ergonomics, Back Compressive
Force, Back Compressive Stress

Copyright 2017 by Bala Kusuma Nanda Kumar Ravilla

Approved by

Dr. Cheryl D. Seals, Chair, Associate Professor of Computer Science and Software Engineering
Dr. Richard Chapman, Associate Professor of Computer Science and Software Engineering
Dr. Richard Sesek, Associate Professor of Industrial and Systems Engineering

ABSTRACT

A better designed work place can prevent up to one-third of the back injuries by enhancing occupational safety. Backster™, a biomechanics-training aide, allowed for experiential ergonomics training in a more engaging way. A companion web application to Backster™ would be a versatile assessment tool that can be utilized independently of the prototype.

The Backster™ web application allows for the personalization of results while providing quick assessments of low back pain risk during lifting tasks. The vision for the Backster™ is to support a user to quickly and easily adjust the lifting and personal parameters such as gender, height, weight, load, and posture. Additionally, Backster™ also enables the ergonomist to assess the capabilities of an entire classroom of students in terms of performing lifting tasks under specific set of conditions. The application supports data visualization to increase ease of analysis.

ACKNOWLEDGEMENTS

I would like to express my appreciation and thanks to my advisor, Dr. Cheryl D. Seals for supporting my graduate studies and research, and for being a great mentor. I would also like to thank my thesis committee members, Dr. Richard Chapman and Dr. Richard Sesek, for reviewing my work and providing valuable feedback as well as for their constant encouragement. I extend my gratitude to my colleagues Allison Holt, Adam Brown, Michael Carbaugh, Sumeet Wilkhu and Amol Joshi who worked alongside me during the initial stages of this research. Finally, I would like to greatly thank my parents, and my dear friends, I couldn't go this far without their help.

TABLE OF CONTENTS

| | |
|--|-----|
| ABSTRACT..... | ii |
| ACKNOWLEDGEMENTS..... | iii |
| CHAPTER 1 INTRODUCTION..... | 1 |
| CHAPTER 2 LITERATURE REVIEW..... | 4 |
| 2.1 Ergonomics..... | 4 |
| 2.2 User Interface Design..... | 5 |
| 2.3 Responsive Web Design..... | 7 |
| 2.4 Scalable Vector Graphics..... | 8 |
| 2.5 Evaluation Techniques..... | 9 |
| CHAPTER 3: METHOD..... | 11 |
| 3.1 Motivation/ Study Problem..... | 11 |
| 3.2 Hypotheses..... | 12 |
| A. Ease of Use:..... | 12 |
| B. Effective feedback:..... | 13 |
| C. Aesthetically Pleasing:..... | 13 |
| D. Efficient code complexity/technologies used:..... | 13 |
| E. Satisfaction:..... | 13 |
| F. Features Provided:..... | 13 |
| 3.3 Specification..... | 13 |
| 3.3.1 Wire Frame..... | 13 |
| 3.3.2 Software and Hardware Requirements..... | 14 |
| 3.4 Conceptual Model..... | 15 |
| 3.4.1 Project Schedule..... | 15 |
| 3.4.2 Lexicon..... | 15 |
| 3.4.3 Functional Requirements..... | 17 |
| 3.4.4 Low-Fidelity Prototype..... | 18 |

| | |
|--|----|
| 3.4.5 Scenarios | 18 |
| CHAPTER 4: IMPLEMENTATION..... | 20 |
| 4.1 Sprint 1..... | 20 |
| 4.1.1 User Interface..... | 20 |
| 4.1.2 Middle Layer..... | 21 |
| 4.1.3 Testing..... | 22 |
| 4.2 Sprint 2..... | 22 |
| 4.2.1 User Interface..... | 22 |
| 4.2.2 Middle Layer..... | 22 |
| 4.2.3 Testing..... | 23 |
| 4.3 Sprint 3..... | 23 |
| 4.3.1 User Interface..... | 23 |
| 4.3.2 Middle Layer..... | 24 |
| 4.3.3 Testing..... | 24 |
| 4.4 Sprint 4..... | 24 |
| 4.4.1 User Interface..... | 24 |
| 4.4.2 Middle Layer..... | 24 |
| 4.4.3 Testing..... | 25 |
| 4.5 Backster™ Wrap-Up..... | 25 |
| 4.6 Implementation Details | 26 |
| 4.7 Phase-2 application | 27 |
| 4.7.1 Improving Visualization through Scalable Vector Graphics | 29 |
| 4.7.2 Storing user data in MongoDB | 32 |
| 4.7.3 Registration and Login implementation..... | 33 |
| 4.7.4 Ergonomist/Client Dashboard..... | 34 |
| CHAPTER 5: EVALUATION/RESULTS | 39 |
| 5.1 Study Details | 39 |
| 5.2 Results before BCF and BCS Scales Implemented..... | 41 |
| 5.3 Results after BCF and BCS Scales Implemented | 43 |
| 5.4 Pre-Survey Results..... | 45 |
| 5.5 Post-Survey Results | 47 |
| 5.6 Analysis of All Responses | 54 |
| 5.7 Advantages of using MeteorJS | 55 |

| | |
|---|----|
| 5.8 Performance Improvements | 56 |
| CHAPTER 6: CONCLUSION AND FUTURE WORK | 60 |
| REFERENCES | 61 |

LIST OF FIGURES

| | |
|--|----|
| Figure 1: Backster™ Hardware Prototype | 2 |
| Figure 2: Backster™ Wire Frame | 14 |
| Figure 3: Backster™ Low-Fidelity Prototype | 18 |
| Figure 4: Filled-In Low-Fidelity Prototype | 19 |
| Figure 5: User Interface Design upon Completion of Sprint 1 | 21 |
| Figure 6: Backster™ Prototype before Calculations..... | 25 |
| Figure 7: Backster™ Prototype after Calculations..... | 26 |
| Figure 8: Backster™ Application Homepage | 28 |
| Figure 9: Color Coding on JPEG Images..... | 29 |
| Figure 10: Backster™ Posture scaling with SVG transformations | 30 |
| Figure 11: Risky Lift color coding with SVG | 31 |
| Figure 12: Backster™ Posture after SVG Implementation | 31 |
| Figure 13: Safe Lift color coding with SVG | 32 |
| Figure 14: Backster Registration Page | 33 |
| Figure 15: Backster Login Page | 33 |
| Figure 16: Dashboard after Login..... | 34 |
| Figure 17: Dashboard with input stats charts | 35 |
| Figure 18: Dashboard with user stats charts | 36 |
| Figure 19: Dashboard with Clear Data and Start/Stop Save | 37 |
| Figure 20: Dashboard input stats charts with AdjustWeight | 38 |
| Figure 21: Dashboard input stats charts without AdjustWeight | 38 |
| Figure 22: Survey result for background..... | 45 |
| Figure 23: Survey result for knowledge on UID | 45 |
| Figure 24: Survey result about Demographics..... | 46 |
| Figure 25: Survey result about browsers | 46 |
| Figure 26: Backster is easy to use | 47 |
| Figure 27: Comfortable using the Backster application..... | 47 |
| Figure 28: Easy to find help/hints | 48 |
| Figure 29: Balance between graphics and text for feedback..... | 48 |
| Figure 30: Backster Calculation make sense | 49 |
| Figure 31: Trust Backster | 49 |
| Figure 32: The user interface was pleasant | 50 |
| Figure 33: Like the design of the application | 50 |
| Figure 34: Graphics are Pleasing | 51 |
| Figure 35: Color scheme is attractive..... | 51 |
| Figure 36: Understand the information application is asking me | 52 |

| | |
|--|----|
| Figure 37: Typography/fonts are easy to read..... | 52 |
| Figure 38: Satisfied with calculations..... | 53 |
| Figure 39: Application Intuitive..... | 53 |
| Figure 40: Confidence in ability to use Backster..... | 54 |
| Figure 41: Backster™ Android app loading screen..... | 55 |
| Figure 42: Backster™ Android app homepage..... | 56 |
| Figure 43: Performance measure of clear data method..... | 57 |
| Figure 44: Performance measure of save data method..... | 57 |
| Figure 45: Performance measure of move data and load data methods..... | 57 |
| Figure 47: Performance comparison between one and two inserts..... | 58 |
| Figure 46: Performance comparison between moveData vs clearData..... | 58 |
| Figure 48: Performance measure of load data method..... | 59 |

CHAPTER 1 INTRODUCTION

Occupational Safety is defined as the wellbeing and health of people employed in a work environment. It is very important at work and focuses on prevention of hazards [1]. Improper safety measures can result in injuries and illnesses to the people. Lapses in occupational safety can prove to be consequential not only to the victim but also to their employer and family. For instance, the employer will have to bear the cost of lost work hours, increased insurance premiums, and deal with possible law suits. On the other hand, the victim's family may have its sole bread winner incapacitated, or worse, lost forever [2].

Dr. Richard Seseck of Auburn University's Samuel Ginn College of Engineering conducts research for the Industrial and Systems Engineering department. The focus of his research centers around occupational safety and ergonomics. While investigating anthropometric factors related to lower back pain, Dr. Seseck developed Backster™. The hardware prototype allowed an individual to position a wooden model in various lifting postures. A scale attached to the model reflects the back-muscle tension associated with each posture. The goal of Backster™ was to train safety officers on manufacturing floors on how to identify and calculate back related risk factors when a worker is performing potentially dangerous lifts. Backster™ was being used in classroom settings since the hardware prototype was relatively expensive in terms of monetary cost and time needed to build and assemble the prototype. The current version of Backster™ can be seen in Figure 1.

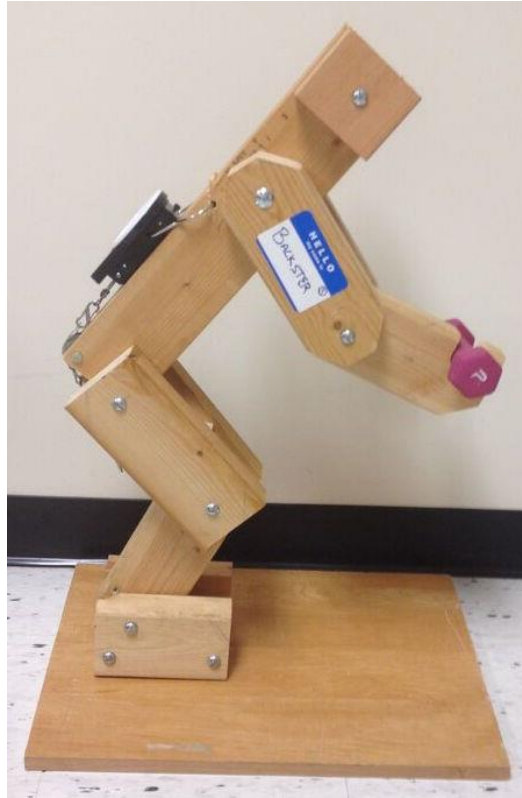


Figure 1: Backster™ Hardware Prototype

Building a physical model of Backster™ is financially unfeasible. Owing to this reason, Dr. Seseek's initial requirements were to expand Backster™'s horizons through the creation of a software prototype. With a software prototype, the initial time investment was high, but, once the initial software prototype was developed, it became easier to disseminate Backster™. The main advantage of creating a virtual version of Backster™ is that, the back compressive force (BCF) values for any individual can be computed based on their physical measurements. With the hardware prototype, a generic result was returned; whereas, with the software version, we calculated specific results based on the user's gender, height, weight, etc. Dr. Seseek and other members of the industrial engineering (IE) team expanded our requirements by stating the need to utilize the software during lectures and with other training sessions. The software was planned to support experiential learning, i.e. a person will enter their gender, weight, height, load, and posture

into the virtual Backster™ to view the corresponding Back Compressive Force (BCF) and subsequent Back Compressive Stress (BCS). Additionally, the user can also compute BCF and BCS values for several what-if conditions such as varying body weights and different loads. A participant can view the results quickly and not have to worry about reading a scale. Also, with Backster™ in a software format, the user may also simultaneously view a personalized BCS and the BCF.

After discussing Backster™ requirements with the design team, the group decided to create a web application as the first iteration of development. Future teams will be tasked to develop mobile applications. Dr. Seseek and IE team also wanted the users to be able to share their results with a group, and that networking capability had been pushed to the second iteration of the product. However, users will have the ability to export their results into a CSV file.

We developed a web based application for Backster™, which meets all the proposed requirements of our Client.

CHAPTER 2 LITERATURE REVIEW

2.1 Ergonomics

The plan for Backster™ was to help individuals understand their limits in terms of lifting a weight via virtual exploration. One of the most common and costly issues in the workplace was occupational back pain and injury. As a general rule, the more complex a model, the more accurate the results. Having to compute these calculations manually increased the likelihood of an error, which potentially led to an individual being cleared to perform a dangerous lift. Loertscher, Merryweather, and Bloswick proposed a revised model for lifting that was more accurate. Their modification involved simplifying assumptions, like reducing the rounding of the torso angle values. These modifications were shown to be more accurate and cost-sufficient for risk evaluation in terms of workstations and lifting techniques [3].

The construction industry is more prone to work-related injuries. Shoubi, Barough, and Rasoulijavaheri researched ergonomic principles to remedy the plague of injuries construction workers have suffered. These Musculoskeletal Disorders (MSDs) have accounted for up to 40 illnesses in countries all over the world. In the United States, MSDs were the most common work-related problems in the construction industry. Ergonomics was considered to be one of the most effective techniques for maximizing work performance while minimizing health risks to employees. Prevention was the most preeminent way to alleviate work-related risks [4]. The Backster™ web application has the ability to take an individual's characteristics and alert employers if the lift was too dangerous for the employee to attempt. Thus, Backster™ has a positive impact in reducing work-related injuries in the construction industry.

Since Backster™ aimed to diminish the number of low back injuries, it was important for the team to also examine studies on low back pain. Thiese et. al., were researchers who focused

on the factors involved with low back pain. In their study, participants came from thirty different facilities where employees performed a variety of manual material handling. According to their findings, pain ratings varied widely, but the location of the pain was mainly focused around the paraspinal region [5]. Backster™ has the potential to lessen the number of employees who experience low back pain due to lifting in these manufacturing facilities.

The ergonomic principles used in Backster™ will impact the number of work-related injuries. The implementation of Backster™ at manufacturing facilities meant that occupational safety managers had the ability to prevent employees from performing risky or dangerous lifts. Thus, lessening a worker's risk for low back pain or Musculoskeletal Disorders.

2.2 User Interface Design

The Backster™ collective also investigated user interface design standards and techniques. The Backster™ web application was designed and developed from scratch, so the team worked with our client in a series of participatory design sessions to gather requirements and guidance from the Industrial Engineering team and Backster™ collective to collaborative plan the initial design of the web-based version of the application.

When designing a user interface, it was important to consider the target audience for the interface. Many countries have laws requiring web applications to ensure accessibility to people with disabilities, functional limitations, or technological limitations. In terms of developing a user interface, accessibility was one of the most strenuous tasks to achieve. There are so many factors to consider with accessibility, so many designers get frustrated. Miñón, Moreno, Martínez, and Abascal proposed a solution to these accessibility problems using a methodological approach based on User Interface Description Language (UIDL) on the User Interface Extensible Markup

Language (UsiXML) framework. The strategy involved the integration of accessibility requirements into design primitives. The proposed approach conformed to the Web Content Accessibility Guidelines (WCAG). Their study concluded the quality of accessibility can be guaranteed by taking advantage of developments in technologies based on UsiXML. In terms of the Backster™ product, the team did not anticipate a necessity for the web application to handle accessibility issues. With it being the first iteration, developing functionality was key and any accessibility issues would be handled in future iterations [6].

Even though accessibility was not addressed during iteration one, Backster™ still needed to reach a large audience. Thus, the team decided to create a web application rather than an operating system specific program. Web design was similar to traditional interface design in that both were highly interactive and incorporated visual elements into the user experience. However, the field of web design possessed a distinct set of challenges. Galitz provided an exhaustive list of design challenges, which was typically viewed as overwhelming. Fortunately, tools exist that bridged the gap between web and interface design, which allowed the designers to focus on a user's needs. The Backster™ team addressed majority of these design challenges by carefully selecting which tools to be utilized during development. All of the developers adhered to a set of coding standards, specifically Google HTML/CSS Style Guide. The development team also relied on the combination of MeteorJS and the Bootstrap framework to structure the project and potentially facilitate responsive design patterns in the future. The knowledge gained from Galitz resulted in easier implementation with a more user-centric approach to designing the Backster™ web application [7].

In summary, there were several considerations taken into account when balancing the customer's requirements with good user interface design.

2.3 Responsive Web Design

Initially, Dr. Seseek and IE team stated a requirement of users accessing Backster™ with their mobile devices. The development team was unable to develop both a website and a mobile application initially, so decided to develop a web application first. The team analyzed alternatives and decided on utilizing responsive web development to provide flexible application access.

Mobile devices and computers are ubiquitous. In the age of the Internet, Baturay and Birtane mentioned that it had become easy to present and access complex systems on virtually all screens with some computational power. Web browsers were the leading candidate in providing a platform to access all the information without the need of enormous amount of memory and power. Responsive web design allows virtually any sized screen to access the responsive applications. The responsive applications were developed to internally handle the adaptation to different screen sizes. When the developers adopted a responsive web design, resolution adjustment, feature enabling/disabling, and quick processing were some of the features that became available. These features played a key role in any instructional web design [8].

Voutilainen, Salonen, and Mikkonen also wrote about creating responsive user interfaces for web pages that were required to be accessible across multiple devices. The average United States consumer typically owned at least two computing devices (a computer and a smartphone) when Voutilainen et al. examined how to implement responsive web designs. The increase of people relying on multiple devices had caused users expectations to be effortlessly switching between their devices whilst interacting with web pages. From a designer's standpoint, the display to the user was required to be optimized based on the device's screen size and not cause the user to have a bad interaction experience. Thus, one of the key issues with responsive web design was how to design a layout that can easily and efficiently converted to different screen sizes. One common solution to this problem was to utilize grids that are based off proportions and not pixel

length. Fortunately, there were various libraries that aid the design of a responsive web page; Bootstrap was one of the most commonly used libraries. Bootstrap was comprised of a flexible grid system. Some responsive libraries forced developers to redesign the entire web page with the specified tools, while others simply encapsulated the existing website. A proper responsive helper reassembles elements that were side-by-side on a desktop screen and stack them on a mobile device. Since the Backster™ developers were designing and developing the software application from scratch and did not have to worry about an existing web page, the usage of the Bootstrap libraries appeared to be the best course of action [9].

As the implementation section will describe, the team decided to utilize the Bootstrap libraries. Thus, a user was not constrained to using a desktop, laptop, or large tablet, even though mobile development was not a priority for the Backster™ development team. Hence, while a responsive web design did not fully meet the customer's ideal Backster™ for smartphones, it did give a starting point for future developers.

2.4 Scalable Vector Graphics

The SVG image format was conceived and developed by the World Wide Web Consortium (W3C) as an XML-based vector image format. In addition to rendering two-dimensional graphics, SVG also supports interactivity and animation. In total, SVG can support three distinct types of graphic objects; Text, Bitmap images, and vector graphic shapes such as paths and outlines. The best applications of SVG are logos, icons, animations, charts, and graphics since this image format is uninfluenced by device type or resolution. More advanced features such as interactive animations can also be created in SVG format by manipulating the SVG XML elements. The same goal can be achieved by accessing the SVG Document Object Model (DOM) through scripting. Scripting in SVG is done by JavaScript while CSS is used for styling [10].

2.5 Evaluation Techniques

Upon the completion of the Backster™, the developers required a technique for evaluating the web application. The developers needed to think about how to evaluate if the functional requirements were met and user satisfaction.

Usability inspection testing was a cost-effective method for evaluating user interfaces at any stage of the software life cycle (i.e. Design, Code, Test & Deployment). Testing was one of the most common techniques for evaluating user interfaces; however, it was also one of the most expensive processes. Therefore, Nielsen declared usability inspection was a widely acceptable alternative to evaluate user interface. There were several usability inspection methods: heuristic evaluation, cognitive walkthroughs, formal usability inspections, pluralistic walkthroughs, feature inspections, consistency inspections, and standards inspections. These provided an alternative way of inspecting and debugging user interfaces [11].

Heuristic evaluation is a usability engineering method for finding the usability problems in a user interface, which allows the assessment to be part of an iterative design process. Heuristic evaluations involved having a small set of evaluators examine the interface and judge its compliance with recognized usability principles. Only after all evaluation had been completed were the evaluators allowed to communicate and their findings aggregated. The delay in communication ensured the independent and unbiased evaluation from each examiner. The output from using the heuristic evaluation method was a list of usability problems in the interface with references to those usability principles that were violated by the design. With the use of Heuristic Evaluations, Industrial Engineering experts were given a set of heuristics to indicate their impressions of the design and indicate the features that they liked and did not like. IE evaluators were not given the ability to provide open ended comments in the heuristic evaluation, but we did provide opportunities for them to provide comments in a follow up retrospective interview about

the initial application. Heuristic evaluation did not provide a systematic way to generate fixes to the usability problems or a way to assess the probable quality of any redesigns. The time constraint for developing Backster™ initial prototype allowed the team to perform a brief heuristic evaluation. The team had to rely on this heuristic evaluation and user satisfaction results to refine the next planned version of the Backster prototype [12].

Another group of researchers discussed the UX Curve. The goal of user experience design in industry was to improve customer satisfaction and loyalty through the utility, ease of use, and pleasure provided in the interaction with a product. The UX Curve aimed at assisting users in retrospectively reporting how and why their experience with a product changed over time. The results suggested the UX Curve method enabled users and researchers to determine the quality of long-term user experience and the influences that improved user experience over time or caused it to deteriorate. A template was developed for user to draw themselves a curve describing how their experiences had evolved over time. In addition to drawing the curve, a user described the factors that improved or decreased a user's experience. This technique was useful for evaluating the long-term user experience and identifying the major experiences that changed users' attitudes towards the product. The UX Curve relied on users' memories, so the data had the potential to not uncover all the important details of experiences. Since the Backster™ web application was invented from scratch, the development team could not evaluate user satisfaction in the long-term. However, this evaluation method was something future developers need to consider [13].

For the Backster application, the developers performed feature testing and conducted a user satisfaction questionnaire.

CHAPTER 3: METHOD

This chapter introduces the research problem, the hypotheses, and the research questions of the study.

3.1 Motivation/ Study Problem

A research study conducted by the University of Maryland found that, annually more than one million back injuries are sustained in the workplace. This study further revealed that, back injuries account for one in every five injuries and illnesses in work environment. Of all the tasks, manual material handling tasks account for 80% of back injuries [14]. According to Oklahoma State University research study, with a better designed work place up to one-third of the back injuries can be prevented. The study stated that, the second most common reason for nonattendance in the general workforce is back injuries [15]. According to U.S. Bureau of Labor Statistics, an average of 12 days is required for the US workers who suffered ergonomic injuries, before returning to work. According to The Center for Construction Research and Training (CPWR), the main causes for work-related back injuries are repeated lifting of materials, whole body vibrations, sudden movements, bending over for extended periods, lifting, and twisting at the same time [16]. According to U.S. Department of Labor, annually \$50 billion is spent to treat back disorders in the US [17].

After researching ergonomic tools, it was found that there only a few available online. The National Institute for Occupational Safety and Health (NIOSH) provides an application called “NIOSH Lifting Equation Calculator application (NLE Calc)” to quickly assess the risks of manual lifting tasks by the users. For calculating manual lifting risks of complex and multiple lifting tasks, NLE Calc uses the Composite Lifting Index (CLI) [18]. Auburn University’s Occupational Safety, Ergonomics, and Injury Prevention Program provides two applications named Lifting Fatigue

Failure Tool (LiFFT) and Distal Upper Extremity Tool (DUET) for assessing the risks in lifting and repetitive upper extremity tasks [19]. Oregon Department of Consumer and Business Services offers Safe Lifting Calculator application [20]. After inspecting/using these applications, it was found that all the applications are for repetitive tasks and do not provide an appealing user interface/experience. The problem is assessing risk for a one-time lifting task, for which there is no application as of now. Ergonomists make these calculations by hand or rely on simple software such as Microsoft Excel.

3.2 Hypotheses

The overall usability of the newly designed application will be evaluated based on ease of use, effectiveness (feedback/design of the user interaction and user interface), aesthetics, and efficiency (code complexity/technologies used). This research is based on usability questions that will be tested at the end of the study. This empirical study focuses on the overall usability of the newly designed application. At the end of the study, the usability analysis of the application by experts and other test participants will be used to test the hypotheses. To evaluate this implementation, a series of studies will be conducted, qualitative evaluation and usability analysis is conducted through user surveys. As there is no previous version/application to compare with, user surveys and expert/client opinion is performed to evaluate the application. The benchmark is a rating of greater than 3 on a 5-point rating scale. The surveys are conducted on user population of 180 users, and the evaluations above the benchmark is considered as a satisfactory level of performance. The areas that are assessed are as follows:

A. Ease of Use:

The overall application design is intended to be perceived as easy to use.

B. Effective feedback:

The application provides an immediate/effective feedback to the user.

C. Aesthetically Pleasing:

The layout and design of the application interface is intended to be aesthetically pleasing.

D. Efficient code complexity/technologies used:

The technologies used, or code complexity is the best approach to this application.

E. Satisfaction:

The overall design of the application is intended to satisfy users.

F. Features Provided:

The overall satisfaction with provided features meet the user/ergonomist/client's expectation.

3.3 Specification

When defining the specifications of the Backster™ web application, the design team found it useful to construct a wire frame and then determine what the software and hardware requirements would be.

3.3.1 Wire Frame

Figure 2 shows the team's wire frame. Since Backster™ was initially being designed as a web application the image in Figure 2 would be what a user would see in his or her web browser. After presenting this material to the client Dr. Seseek, the team decided to add in a toggle button to hold the option to export the data to a CSV file. The toggle button would be located in the top banner on the left side. The rest of the banner held the name of the web application. The image on the left of the frame helped give the user a visual of what the parameters on the right represent. This visual tool should help decrease any user error with him or her not understanding what was

required to calculate the BCF or BCS. Once the parameters had been entered, the user would hit the calculate button to display the results. The user's BCF and BCS values would be displayed on the bottom of the left image superimposed on a scale. The scale's coloring would represent if the lift was potentially safe (green), risky (yellow), or dangerous (red). The bottom of the webpage had the copyright information for the product along with a way to contact the Backster team lead.

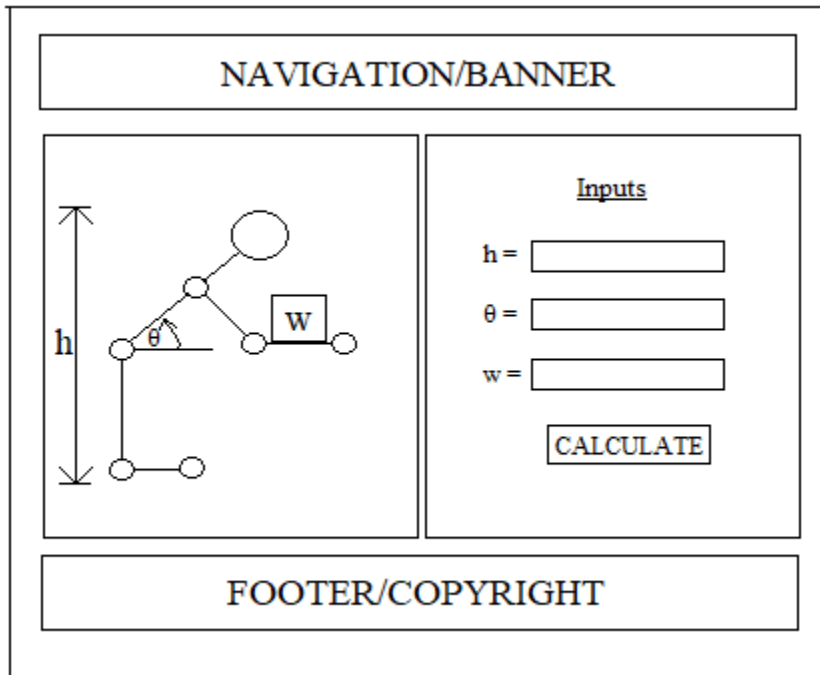


Figure 2: Backster™ Wire Frame

3.3.2 Software and Hardware Requirements

Given the wire frame above, it was decided the only software requirement for a user to take advantage of the Backster™ web application will be a modern web browser. As for hardware requirements, the user will require a desktop computer, laptop, or a tablet to access the web application and no run into any difficulties. Since the smartphone development was postponed to the next iteration, the user interface design team was not concerned with how the web application will function on small mobile devices, see Future Work (Section 6).

3.4 Conceptual Model

The conceptual model specified the project schedule, lexicon, software development process, functional requirements, and low-fidelity prototype, and scenarios.

3.4.1 Project Schedule

The Backster™ user interface design team was allotted six weeks to complete the client's request for a web application. The first week was spent selecting which user interface design project to tackle. Then the team created a wire frame to show to Dr. Sese (see Figure 2). During the second week, the details of the conceptual model were mapped out. This process allowed the team to begin developing the software prototype. Week three consisted of compiling the initial project paper and presentation. The fourth week was when the user interface design would be tested in terms of usability and evaluated with rubrics of the team's design. The last two weeks were saved for the completion of the project paper and presentation. This was also the time for corrections to be made to the user interface design based on the feedback from week four.

3.4.2 Lexicon

The Backster™ web application lexicon was comprised of the following terms: Back Compressive Force (BCF), Back Compressive Stress (BCS), NIOSH Scale, Backster™, bending angle/posture, load, height, weight, and distance (from load object). Below are the terms with their definitions with respect to the Backster™ web application:

- Back Compressive Force (BCF) – BCF was a numerical value and used to evaluate the risk of low back injury for a one-time lifting task based on lumbar disc compression. According to Dr. Sese's index, the BCF scale values are as follows: 0 – 770 meant the lift was safe; 770 – 1430 meant the lift was risky; 1430 – 2000 meant the lift was dangerous; and anything above 2000 meant the lift was extremely dangerous to the individual. From the group's understanding the BCF equation was aimed at being an improvement of the NIOSH Lifting Equation.

- Back Compressive Stress (BCS) – BCS was another numerical value based on the result of the BCF. BCS also had potential for improving the predictability of biomechanical models. According to Dr. Seseek’s index, the BCS scale values are as follows: 0 – 400 meant the lift was safe; 400 – 750 meant the lift was risky; 750 – 1000 meant the lift was dangerous; and anything above 1000 meant the lift was extremely dangerous to the individual.
- NIOSH Scale – The National Institute for Occupational Safety and Health (NIOSH) provided national and world leadership to prevent workplace illnesses and injuries. NIOSH offered a lifting equation to provide guidelines for evaluating two-handed manual lifting tasks. Based on an individual’s characteristics, the NIOSH lifting equation recommended a weight limit for an individual to avoid the potential development of lower back pain. This scale of safe and harmful weight was used as background knowledge; Dr. Seseek gave the team different scales for the Backster™ BCF and BCS values.
- Backster™ - The physical Backster™ prototype was comprised of wooden planks and a tension scale measuring pounds (see Figure 1). Based on Backster™’s posture, the scale showed the back-muscle tension for the lift. Note the physical device did not take into account a person’s height or weight, so the value produced represented a generic lift. It also only measured the back-muscle tension, which is the biggest contributor to back compressive force, but does not represent total force on the back.
- Bending angle/posture – The posture represented a parameter for the Backster™ web application. The angle was described as being measure from the vertical. The possible postures were standing upright (angle of 0°), bent a quarter of the way over (angle of 22.5°), bent halfway over (angle of 45°), bent three-quarters of the way over (angle of 67.5°), and bent all the way over meaning the person’s torso was parallel to the ground (angle of 90°).

- Load – In terms of the Backster™ web application, load represented the weight of the object lifted by an individual. Load was measured in pounds.
- Height – Height represented an individual's height. For the BCF calculation, the height needed to be in inches.
- Weight – Weight represented an individual's body weight. For the BCF calculation, the weight needed to be in pounds.
- Distance – Distance referred to the distance between the center of the load and in individual's hip. The numerical value was required to be entered in inches.

3.4.3 Functional Requirements

After getting the basic project description from Dr. Sesek, the following items were deemed functional requirements for the Backster™ web application:

1. User should be able to enter/change body weight.
2. User should be able to enter/change load weight.
3. User should be able to enter/change height.
4. User should be able to enter/change gender.
5. User should be able to enter/change distance.
6. Given the entered data, the program should be able to calculate the Back Compressive Force.
7. Given the entered data, the program should be able to calculate the Back Compressive Stress.
8. Based on the results calculated, the result boxes will be color-coded (green – good, yellow – moderate, red – problematic).

9. User should be able to export the parameters entered and the calculated results into a CSV file.

3.4.4 Low-Fidelity Prototype

A low-fidelity prototype was created to ensure the client's expectations were being met. Since the prototype of the web application did not need to be fully functional, the image was created using Photoshop. Figure 3 shows the low-fidelity prototype of the Backster™ web application with the inputs blank.

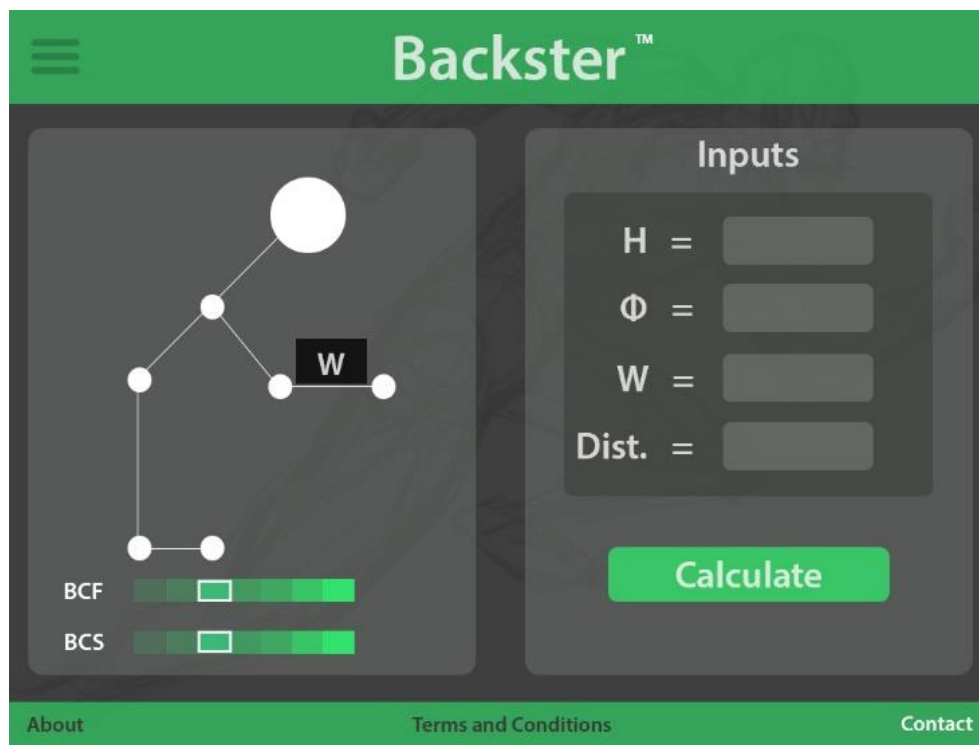


Figure 3: Backster™ Low-Fidelity Prototype

3.4.5 Scenarios

In order to fully understand how the Backster™ web application needed to function, two scenarios were developed in relation to the prototype. Two persons were created: Jim and Jane.

In the first scenario, “Jim” visited the Backster™ website. He filled in the personal information: gender – male, height – 69 inches, and weight – 180 pounds. Jim selected a posture

image of 22.5 degrees, i.e. Backster™ was bent a quarter way down. He entered in a load of 25 pounds and a distance of 20 inches. Jim clicked the calculate button. Jim viewed his BCF and BCS scores on the scale. With a BCF of 519.76 pounds and BCS of 208.12 pounds per square inch, the lift was shown as being safe for a man of John’s size. Figure 4 shows what the low-fidelity prototype looks like with the first scenario.

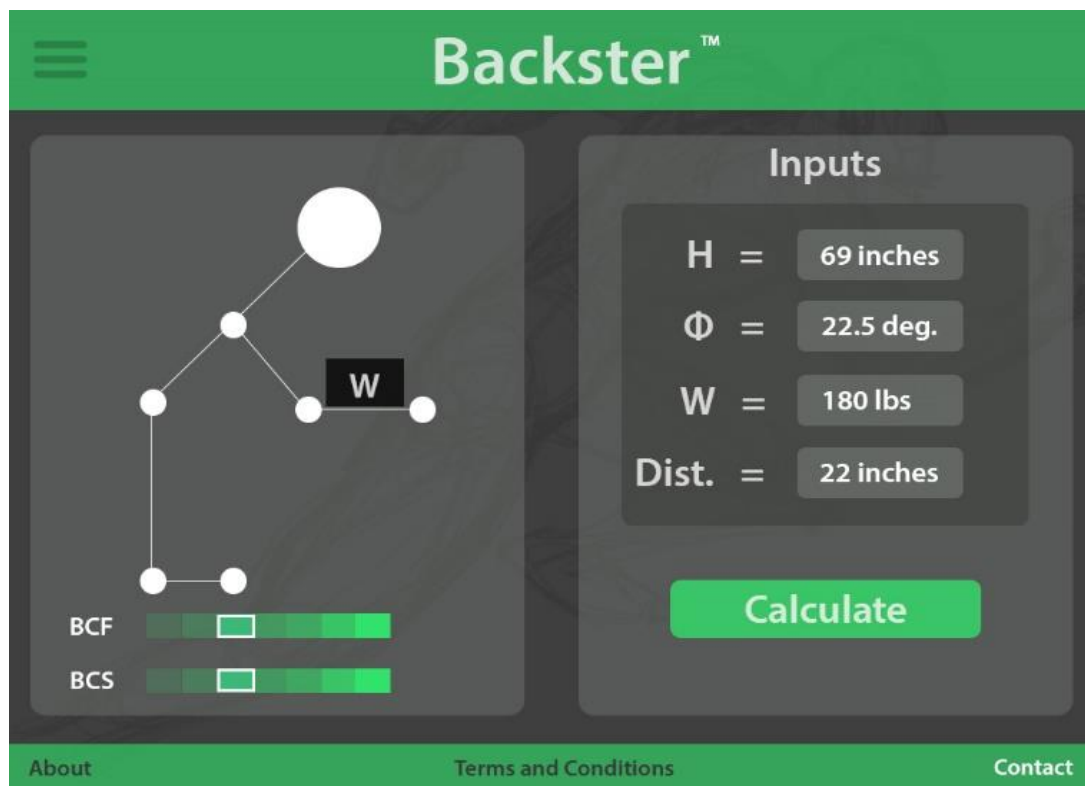


Figure 4: Filled-In Low-Fidelity Prototype

In scenario two, “Jane” visited the Backster™ website. Jane calculated her BCF and BCS. Jane clicked an export button. Jane received a test file via download including calculation parameters and the results.

CHAPTER 4: IMPLEMENTATION

This section describes how the Backster™ web application was created. Since the software development process was Scrum, the section is divided into sprints. Scrum is an agile software development framework which is iterative and incremental for managing software development. It is very useful in cases like new software product development, in which there are unclear requirements or emerging requirements. The Scrum process allows to adapt to these requirement changes flexibly. It also allows to develop a working prototype as quickly as possible so that the customer can offer their suggestion or additional direction which is crucial for a new software product development [21]. The design team was divided into three committees: User Interface, Middle Layer, and Testing. Hence, within each sprint, a summary is provided from each committee. After reviewing all the sprints, the implementation details in terms of software languages and tools were discussed.

4.1 Sprint 1

Sprint 1 began after the project had been selected and Dr. Sesek, the client, had approved the wire frame. At this point, Dr. Sesek provided the team with the formulas to calculate BCF and BCS.

4.1.1 User Interface

The user interface design team created an HTML mockup of the low-fidelity prototype. After discussing implementation with the client, the Backster™ design team decided to have the posture images represent the bent angle parameter. The user would have to click through the various postures to select the bent angle for the BCF and BCS calculation. Thus, during the first sprint, the five posture images were created. The images would be added to the HTML user

interface design during Sprint 2. Figure 5 shows the user interface design at the conclusion of Sprint 1.

4.1.2 Middle Layer

The team decided to use Meteor JavaScript for developing the capabilities necessary for the Backster™ web application to function properly. During the first sprint, the middle layer team began learning how to utilize MeteorJS. They developed the CSV generation technique that allows a user to export his or her BCF and BCS results along with the parameters that led to those results. In order to test the CSV generator, a mocked-up user interface was created. Then, a GitHub account was setup and a file structure was selected for the project. One group member tackled the BCF calculation while the other handled the BCS calculation. The two calculations were then merged together, and errors were fixed. By the end of Sprint 1, the middle layer was ready to be merged with the user interface.

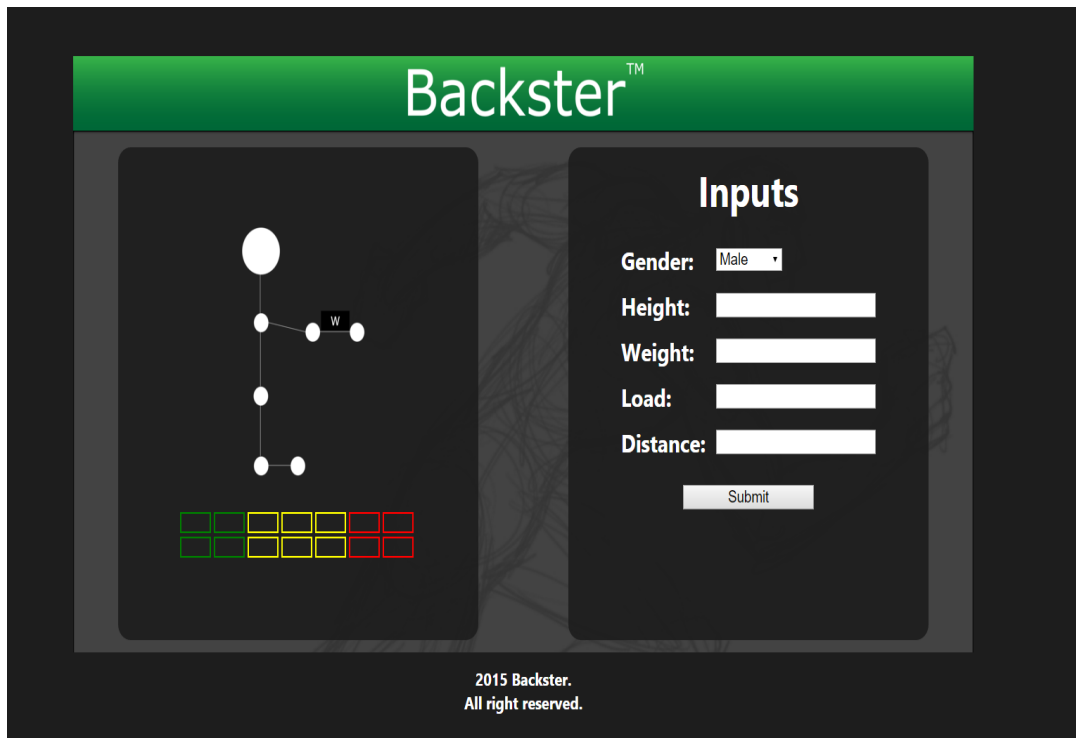


Figure 5: User Interface Design upon Completion of Sprint 1

4.1.3 Testing

Since the web application had not been fully developed, the testing committee created a test plan document. This Excel spreadsheet contains possible test cases.

4.2 Sprint 2

At the beginning of Sprint 2, Dr. Seseok approved the initial user interface design and provided more resources for the designers to examine while developing.

4.2.1 User Interface

During Sprint 2, the interface team started working on the scales that display the BCF and BCS values. The client specified he wanted the scale to contain the colors green, yellow, and red since most users of Backster™ would associate those colors with good, cautious, and bad. Along with displaying a correlating color, the web application would also display the numerical value to the user. Another member of the user interface team focused on the manufacturing of the posture selection. Since the images were created in Sprint 1, the team needed to figure out how to display the possible Backster™ postures. Ultimately, the group chose to utilize a carousel viewer.

4.2.2 Middle Layer

The middle layer team worked with the user interface team to stitch the JavaScript calculations with the user interface design. In other words, they molded the front-end and back-end together to allow a fully functional experience for a user looking to calculate their own BCF and BCS.

4.2.3 Testing

In order to better test the Backster™ web application, the testing team reviewed the documentation for Meteor and Velocity. Being familiar with the software was important for creating a test plan and test cases.

4.3 Sprint 3

By the end of Sprint 3, the team had completed a high-fidelity Backster™ prototype. After getting the customer's approval, the team decided to implement the web application with the Bootstrap libraries.

4.3.1 User Interface

The user interface design committee ported the existing Backster™ prototype elements into a new web design that utilized the Bootstrap libraries. As mentioned in the Background section, the Bootstrap libraries allowed developers to create responsive web applications. Thus, Backster™ was accessible to those on smartphones. The design team did not focus on the design of the mobile version. Hence, the team only ensured the two major elements of Backster™, as seen in Figure 6, shifted to being stacked when accessed on a mobile device instead of being shown side-by-side.

The Backster™ team was more concerned with implementing functionality, so they agreed to modify the planned color scheme to match the default Bootstrap coloring. This decision sped up the development of the functional prototype. The customer did not mind the change because he, too, wanted to have a working online form of Backster™.

In the low-fidelity prototype shown in Figure 3, the user was expected to enter his or her height in inches. Making users convert their height from feet and inches to just inches had the potential to lead to miscalculations. In order to make the product more user-friendly, the design

team changed the height input. Figure 6 showed the height input was a drop-down menu that separated feet and inches.

The final change to the user interface during Sprint 3 was the posture carousel. Instead of having the user guess the angle of the lift, the Backster™ prototype offered five posture images. When a user selects the calculate button, the team wanted the web application to choose the bending angle based on the posture image visible in the carousel. The different posture images gave users a better way of determining their bending angles without having to do mental calculations or estimations.

4.3.2 Middle Layer

Since Sprint 3's focus was on utilizing the Bootstrap libraries, the middle layer development did not have to make any changes to the code developed in Sprint 2.

4.3.3 Testing

The fully functional prototype was not completed during Sprint 3, so no testing took place.

4.4 Sprint 4

The focus of Sprint 4 was to ensure all of the customer's requirements were met in the Bootstrap version of Backster™.

4.4.1 User Interface

The customer had mentioned the desire to have the Backster™ image look as human as possible. Therefore, the user interface team took time to update the posture images to reflect standard human portions and mirror a simple skeletal image of a man.

4.4.2 Middle Layer

The posture carousel was implemented in the user interface at the end of Sprint 3. Thus, during Sprint 4, the middle layer development team updated the code to read the posture/bending

angle value from the carousel. This action eliminated any need for an angle entry under the personal characteristics.

4.4.3 Testing

The Backster™ tester rigorously examined the prototype for potential bugs. While no issues were uncovered with the calculations, he noticed an issue with the calculate button being enabled when all the forms were not completed. In other words, a confused user had the ability to calculate the BCF and BCS values for a lift without giving the weight of the load. After discussing his findings with the entire team, the tester implemented input validation on the form. With the final version of the Backster™ prototype, a user had to provide all information before being able to calculate the force and stress of a lift.



Figure 6: Backster™ Prototype before Calculations

4.5 Backster™ Wrap-Up

Due to team miscommunications, the Backster™ web application was not finished at the end of the fourth sprint. The day after the end of the sprint, the team wrapped up loose ends. Due to some technical issues, not all of the functionality was present in Sprint 4's Backster™. For

instance, the scale at the bottom of the web application was implemented after then end of the sprint. Upon completion, the scale displayed the color-code for how safe the lift was for an individual (green – safe, yellow – risky, red – dangerous). The developers also fixed any bugs discovered during testing. Figure 6 showed what the user views when they visit the Backster™ website. Figure 7 showed an example of results given an individual’s characteristics, the load weight, the distance the load was from the individual, and the individual’s posture during the lift.

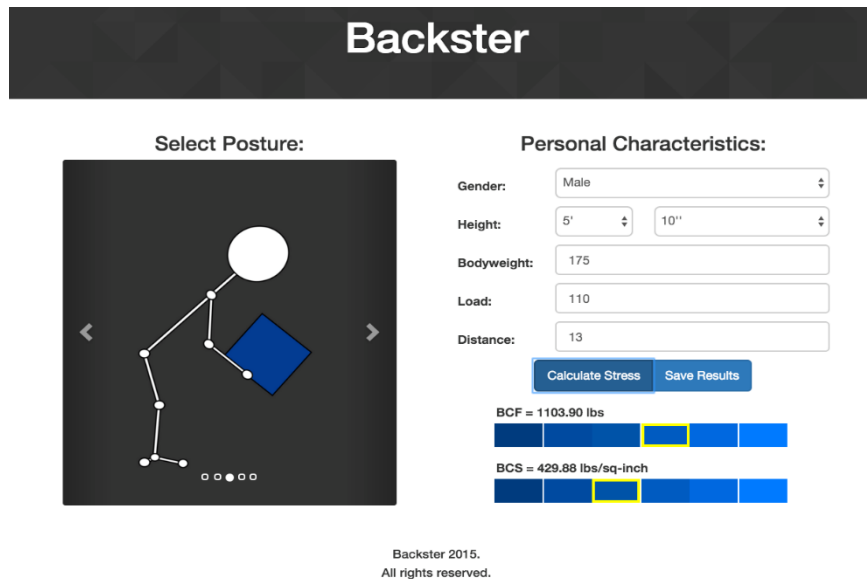


Figure 7: Backster™ Prototype after Calculations

4.6 Implementation Details

The Backster™ team used a plethora of tools to implement the fully functional prototype. The languages used during development were HTML, CSS, and JavaScript. In order to make the web application adaptable to various screen sizes, the developers utilized the Bootstrap libraries. MeteorJS took JavaScript code and simplified the process of designing a web application. The Meteor webpage also hosted the Backster™ web application. For future development, MeteorJS also allowed developers to deploy products onto multiple platforms. This feature was utilized during the user satisfaction questionnaire. The final tool used by the team was GitHub. GitHub

provided a central repository for the web application and developers created branches while creating different features to avoid damaging the working version.

The finalized Backster™ prototype was accessible to individuals with desktops, laptops, tablets, or smartphones given their devices had a modern web browser, like Google Chrome, Safari, Firefox, or Internet Explorer.

4.7 Phase-2 application

Even though the initial version of Backster, provided the users with the required information based on the inputs given. After analyzing it further, a few improvements will make the application even better. The new features implemented are:

1. Color coding the result on the posture image
2. Fixed the Backster equations and calculations
3. New posture image based on ratios
4. Resizing the posture/load based on input height, weight, and load
5. BCS and BCF scales color changes
6. Storing the user details
7. Registration feature for user/admin
8. Login feature
9. Ergonomist/Client dashboard
10. Pie charts with user data
11. Pie charts with ergonomist input combined with user data
12. Clear, Move, and Load data options
13. Start/Stop saving data in the database

The result for the inputs given seems to be slightly deviating from the real world. So, the limits for the safe, moderate and risky limits were recalculated and changed as per Dr. Seseek's comments. The posture image body length ratios were not accurately depicting a real-world human. The posture body lengths are changed according to the ratios given by Dr. Seseek. The values of BCS and BCF are not cleared after the input values are changed. The jQuery code is added to clear the values of BCS and BCF once the input values are changed, the same feature is applied for the color coding as well. The initial BCF and BCS scales were giving the feedback but the users were kind of confused with those scales. So, the new color difference in the scales will clearly provide the users a proper feedback. Figure 8 shows the application homepage after phase-2 implementation.

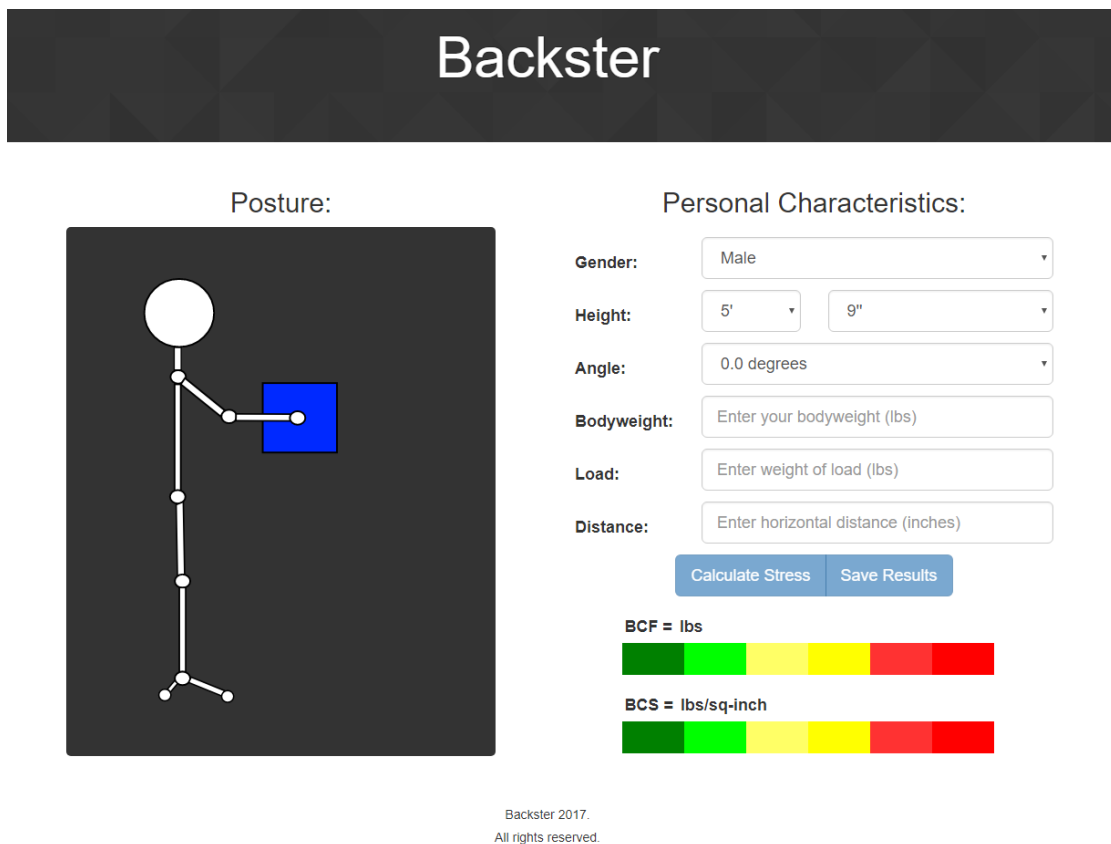


Figure 8: Backster™ Application Homepage

4.7.1 Improving Visualization through Scalable Vector Graphics

After implementing the BCS and BCF scales, they provide the user a feedback whether it is safe or risky to lift a load. It will be effective if the feedback can be represented on the image itself. Preliminary research revealed that, to color code the feedback on the posture diagram, at least 40 JPEG images would be required. Figure 9 shows the images needed to implement color coding using JPEG. After researching online, we found an effective solution that to use Scalable Vector Graphics (SVG) instead of JPEG. This will reduce the number of required images to one. The SVG image format supports transformations and scaling which can be used to resize the SVG elements based on the inputs given by the user.

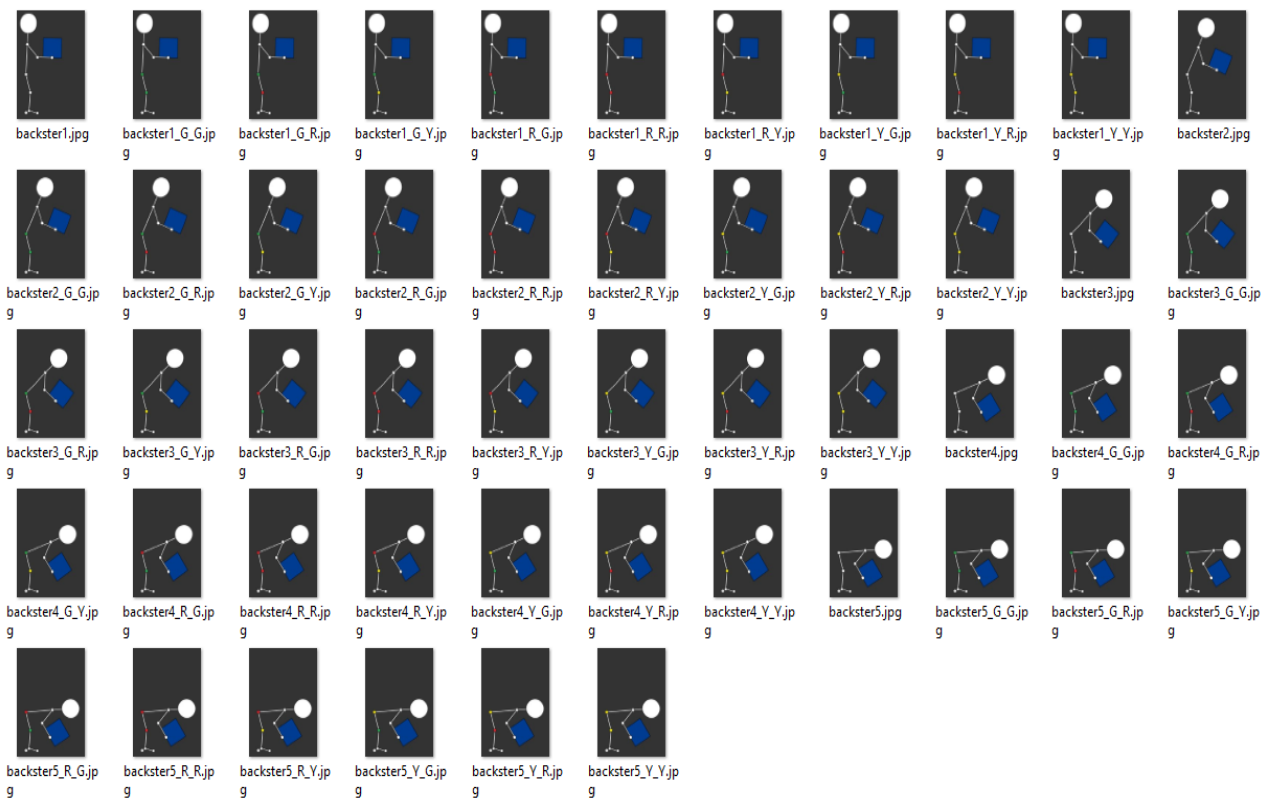


Figure 9: Color Coding on JPEG Images

The color coding was initially implemented by using JPEG images (see figure 9), later on used SVG for efficiency. The height, width of the body is resized based on the inputs from the

user. The Load also resizes based on the input. Figure 10 shows the transformations using SVG. Figure 11 show the posture image implementation using SVG. Once the user inputs all the data and clicks calculate/Enter button the BCS and BCF values are calculated and the corresponding scale is highlighted in the BCS and BCF scales. The color coding is applied to the body and load of the posture image according to the calculated result. Figure 12 and 13 shows the color coding of posture with SVG.

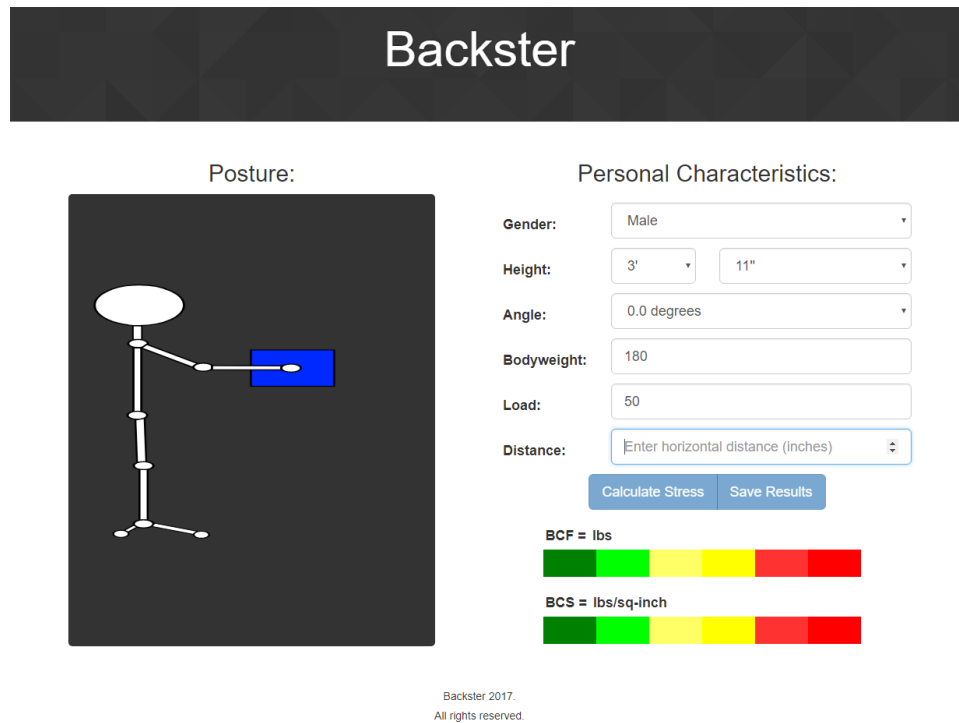
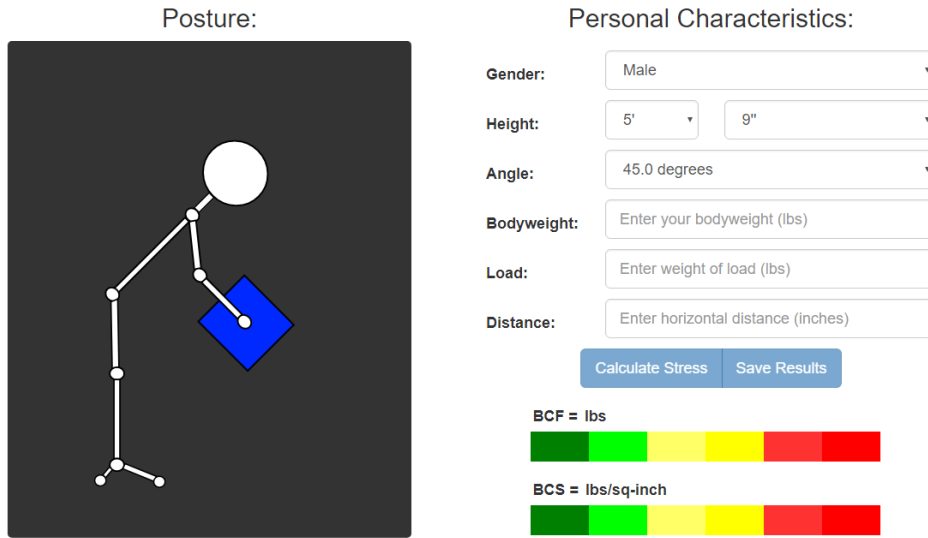


Figure 10: Backster™ Posture scaling with SVG transformations

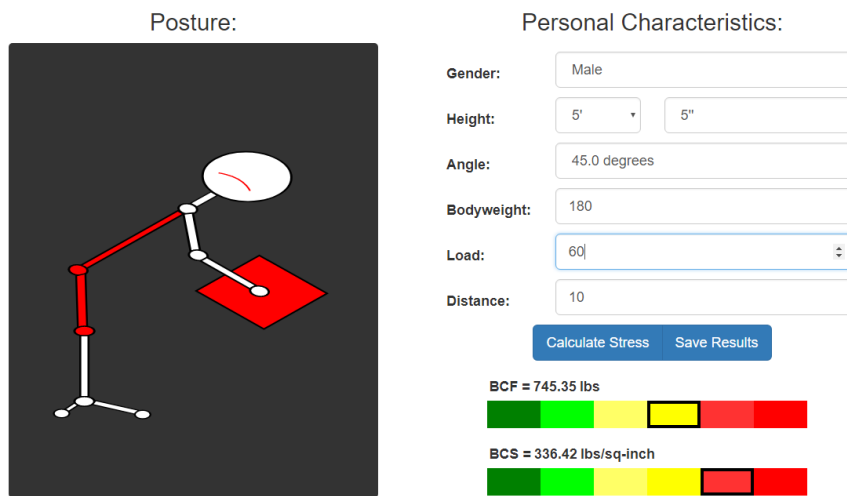
Backster



Backster 2017.
All rights reserved.

Figure 12: Backster™ Posture after SVG Implementation

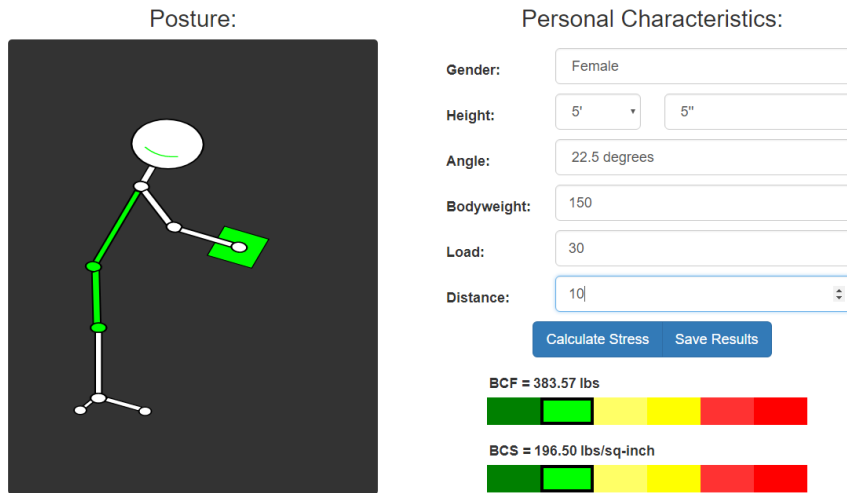
Backster



Backster 2017.
All rights reserved.

Figure 11: Risky Lift color coding with SVG

Backster



Backster 2017.
All rights reserved.

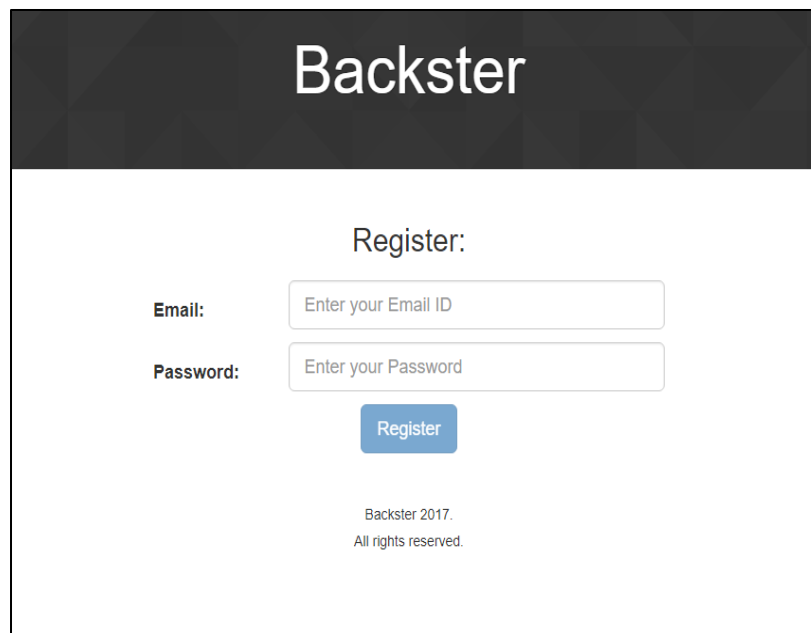
Figure 13: Safe Lift color coding with SVG

4.7.2 Storing user data in MongoDB

The users can input their details and know the result of that lift, but that data was not stored in the initial version developed. So, to store the user data we need a database. After analyzing the databases available, for this purpose using NoSQL databases is better than the SQL databases. Meteor supports MongoDB NoSQL database very well. When the user enters the details and clicks calculate stress, the user input is passed to the save data method on the server side as JSON data. The server-side method stores the data into the MongoDB collection. The MongoDB collection is created with validator that validates and accepts only if the input is valid. If the data is invalid the error message is given to the client-side calling method for logging it.

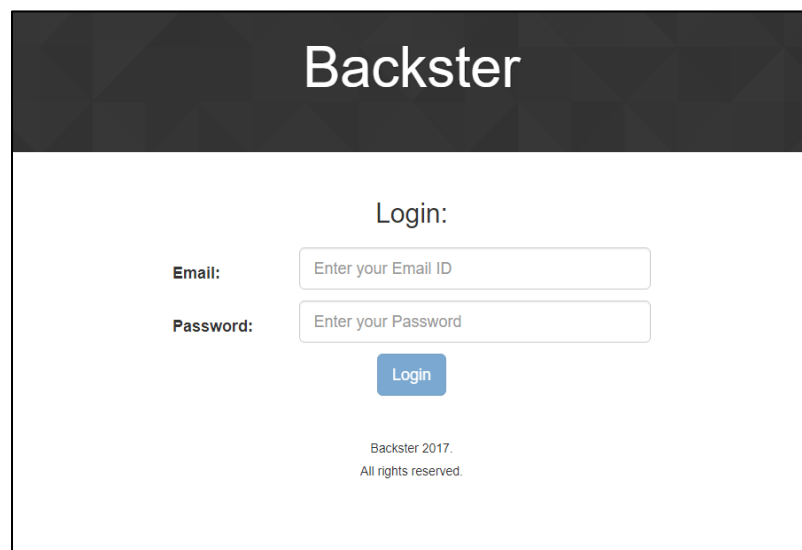
4.7.3 Registration and Login implementation

The registration and login features provide users/ergonomists to create a user profile in the Backster database and let the user/ergonomist to login in to the application. Once the Ergonomist/client logs in to the application, it takes them to the dashboard where they can visualize the user data or the ergonomist's input combined with the user input stored in the MongoDB collection. Figure 14 shows the Registration page and Figure 15 shows the Login page.



The screenshot shows the registration page for Backster. At the top, the word "Backster" is displayed in white on a dark grey background. Below this, the word "Register:" is centered. There are two input fields: "Email:" with the placeholder text "Enter your Email ID" and "Password:" with the placeholder text "Enter your Password". A blue "Register" button is positioned below the password field. At the bottom, the text "Backster 2017. All rights reserved." is displayed.

Figure 14: Backster Registration Page

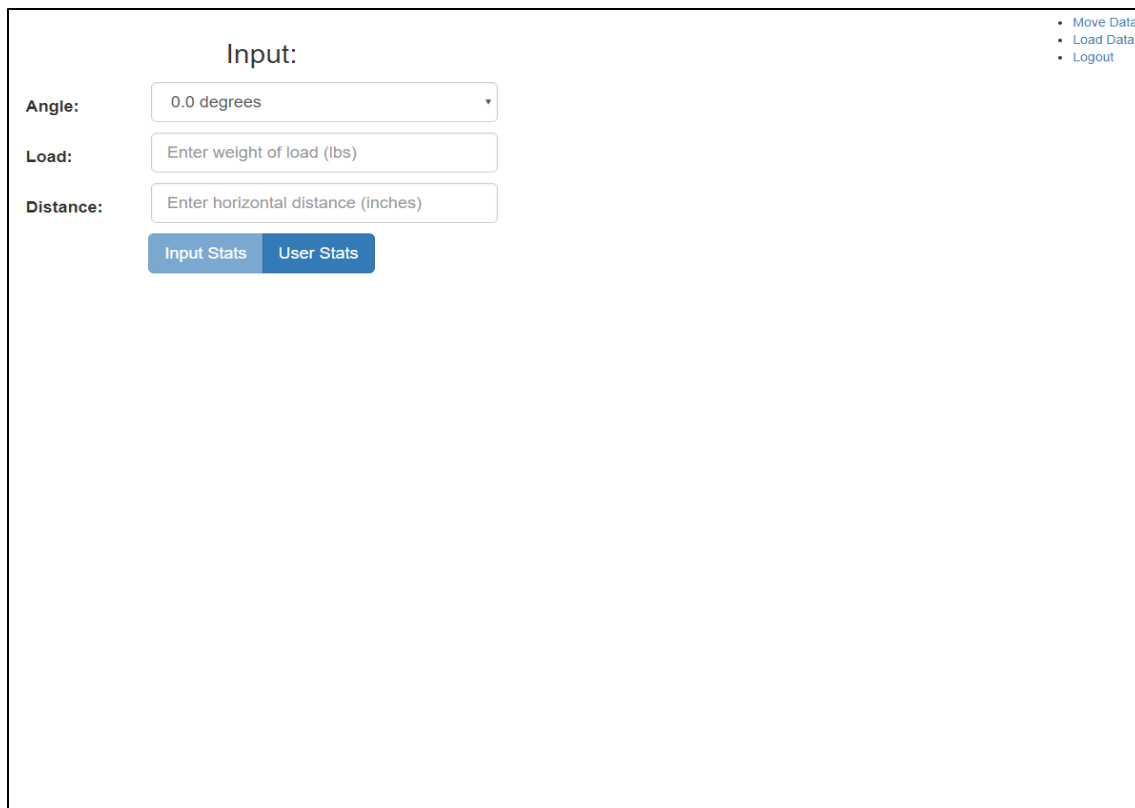


The screenshot shows the login page for Backster. At the top, the word "Backster" is displayed in white on a dark grey background. Below this, the word "Login:" is centered. There are two input fields: "Email:" with the placeholder text "Enter your Email ID" and "Password:" with the placeholder text "Enter your Password". A blue "Login" button is positioned below the password field. At the bottom, the text "Backster 2017. All rights reserved." is displayed.

Figure 15: Backster Login Page

4.7.4 Ergonomist/Client Dashboard

The Ergonomist/Client is provided with the dashboard after successful authentication at the login page. The dashboard gives options to plot pie charts either with the user data or with the ergonomist/client input combined with the user data which is stored in the MongoDB collection. Figure 16 shows the dashboard page after successful login. When the options are clicked, the function call is made for the server-side method get Stats. The get Stats method, pulls the user data in the MongoDB collection, and calculate the BCS and BCF values, determines the result of the lift for each user data. It creates stats, based on each result for the user data and the gender of the user. It returns the result that gives the information of results for all the users and for each gender. Then the pie charts are drawn using this information from the server. The server does not send any user data to the client side, only the results are sent to the client side.



Input:

Angle: 0.0 degrees

Load: Enter weight of load (lbs)

Distance: Enter horizontal distance (Inches)

Input Stats User Stats

- Move Data
- Load Data
- Logout

Figure 16: Dashboard after Login

When the ergonomist/client wants to plot the pie charts, the charts depict the user information from the MongoDB collection. If the ergonomist/client wants to draw charts for a new user data which is from seminar/class users, it is not possible. So, the load and move data options are provided in the dashboard. The move data method is implemented on the server-side to move all the user data to a different MongoDB collection. Now, the ergonomist/client can ask the users to input their data and plot the charts that depicts the current users of the seminar/class. Figure 17 and 18 shows the input stats and user stats pie charts.

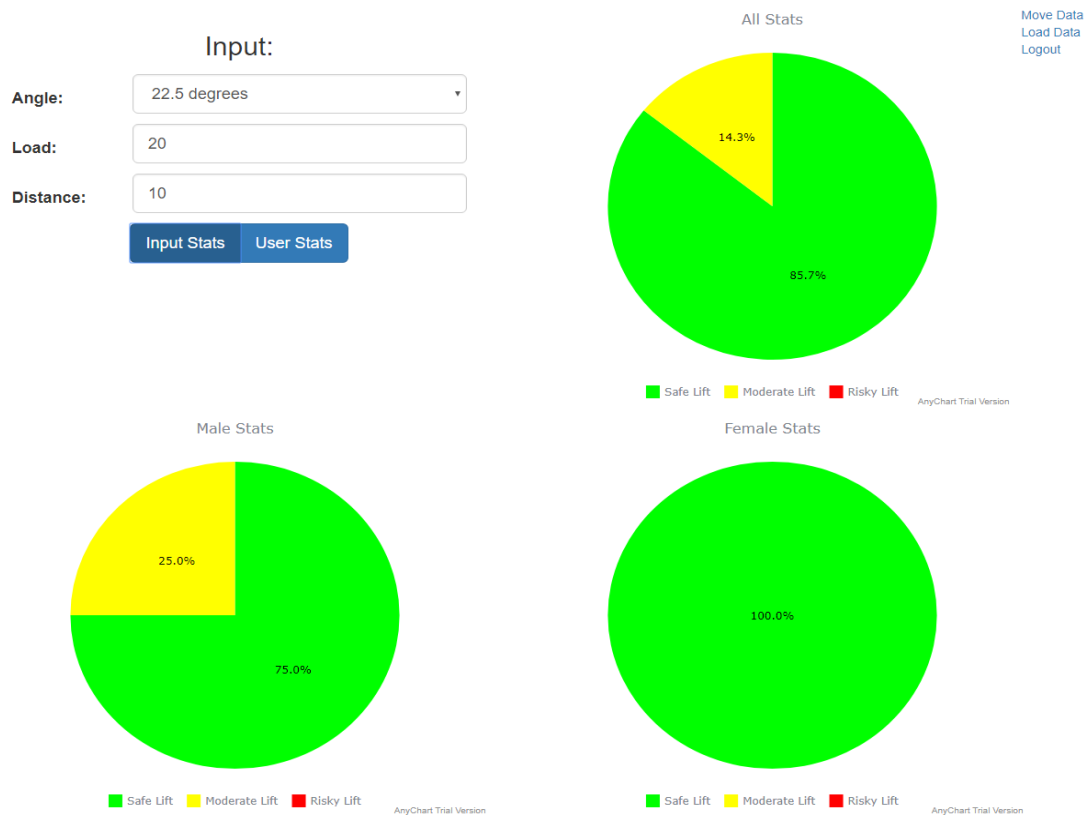


Figure 17: Dashboard with input stats charts



Figure 18: Dashboard with user stats charts

The move data option provided in the dashboard seems to be taking more time to move the data. So, to improve the performance of the application instead of moving data, a clear data option was provided. Now when the user data is sent to the server, it is stored in both the collections (current and all data). So, we can clear the data from current data collection as that data is already present in all data collection. The current data collection is the one used to plot the pie charts. The all data collection is used to store all the user data gathered till now. With the clear data option, the data is cleared immediately, and the performance of the application is improved. The Start/Stop save option is implemented so that storing of the user data in the MongoDB can be controlled. Figure 19 shows the new dashboard with Clear Data and Start/Stop Save options. Also, to improve the performance of load, we implemented a mechanism to switch between the collections instead of loading data from All data collection to Current data collection.

The dashboard interface consists of the following elements:

- Angle:** A dropdown menu currently showing "0.0 degrees".
- Load:** A text input field with the placeholder text "Enter weight of load (lbs)".
- Distance:** A text input field with the placeholder text "Enter horizontal distance (inches)".
- AdjustWeight:** A text input field with the placeholder text "Adjust User bodyweight (lbs)".
- Buttons:** Two buttons labeled "Input Stats" and "User Stats" are positioned below the AdjustWeight field.
- Menu:** A vertical list of links in the top right corner: "Clear Data", "Start Save", "Load Data", and "Logout".

Figure 19: Dashboard with Clear Data and Start/Stop Save

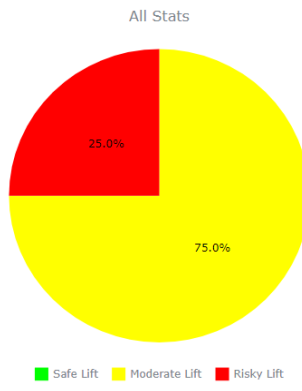
The dashboard only had angle, load, and distance inputs. So, the weight of the user data cannot be adjusted. It will be a great feature if adjusting the user weight is available. So, that the stats can be plotted with decreasing/increasing the users' weights. The value provided in the adjust weight field is added/subtracted to each user's weight while calculating the stats. Figure 20 shows the pie charts without modifying the users weight as the adjust weight field has value 0. Figure 21 shows the pie charts with users' weights increased by 20 lbs. as the adjust weight field has value 20.

Angle:

Load:

Distance:

AdjustWeight:



- Clear Data
- Stop Save
- Load Data
- Logout

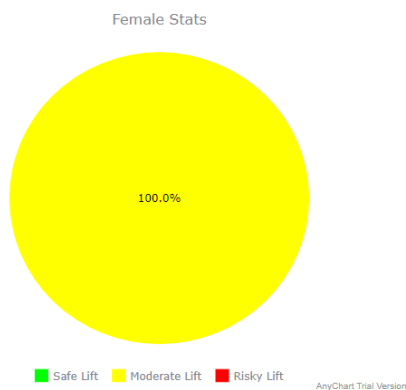
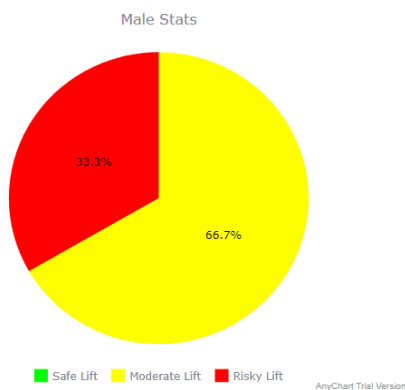


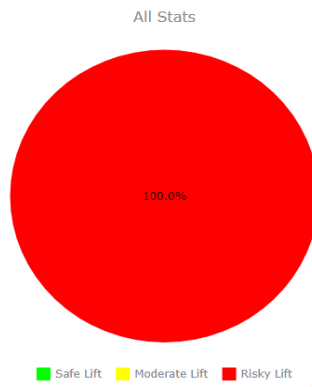
Figure 21: Dashboard input stats charts without AdjustWeight

Angle:

Load:

Distance:

AdjustWeight:



- Clear Data
- Stop Save
- Load Data
- Logout

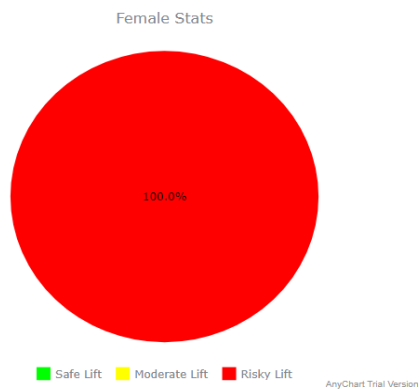
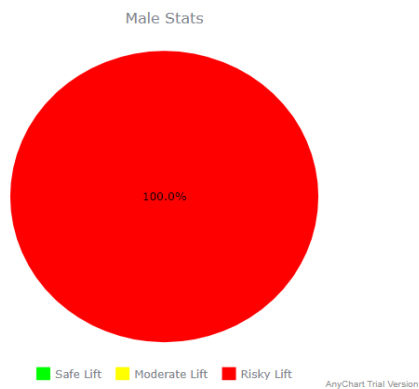


Figure 20: Dashboard input stats charts with AdjustWeight

CHAPTER 5: EVALUATION/RESULTS

In order to evaluate user satisfaction with Backster™, the Backster™ team created a questionnaire to be distributed among students with user interface design and/or ergonomic backgrounds. The group realized the tool needed to be analyzed from both viewpoints.

The questionnaire was designed to analyze a user's level of satisfaction with the Backster™ web application along with a small investigation into usability testing. Along with making sure the user had a satisfactory experience, the group wanted to examine the user's performance with the web application. A portion of the study was conducted in the controlled environment of a classroom, while the other portion was more of a field study. The individuals participating in the field study filled out and submit their questionnaires from their own personal computers in their labs or homes.

The questionnaire results were separated into two groups. The first group participated in the survey in a controlled environment. The version of Backster™ they tested did not have the BCF and BCS scale at the bottom of the screen. The second group took the survey in either their labs or homes and were able to see the product with the scale. Due to the difference, the development team decided to analyze the results separately and combine them after examination. The team used descriptive statistics to assess the results.

5.1 Study Details

The study began with participants answering some preliminary questions about their background in terms of ergonomics, occupational safety, and user interface design. The team also saw it as important to note if the participant was a student or an individual working out in industry. Since the focus of the first Backster™ iteration was on web application, the group also wanted to know what web browser(s) participants commonly used.

After completing the pre-survey, participants used the following three scenarios to interact with the web application:

1. Sally is 62 inches tall and weighs approximately 146 pounds. In her job, she spends roughly half of her day moving light boxes, weighing roughly 10 pounds, from 1 conveyer belt to another. This causes her to lift the box with a slight bend in her back (about 1/4 of the way between her vertical posture and a completely bent posture). Calculate her BCF and BCS for a typical lift at work.
 - a. Select Female for Sally's gender.
 - b. Enter 62 into the inches input.
 - c. Enter 146 pounds into the weight input.
 - d. Enter 10 pounds for the load input.
 - e. Adding a presumed horizontal distance.
 - f. Press the Calculate button.
2. Ben is 72 inches tall and weighs approximately 250 pounds. He frequently changes tires on the job at Pep Boys. Each tire weighs approximately 50 pounds, and Ben is completely bent over (his upper body is horizontal to the ground) while he lifts the tire onto the hub. A presumed horizontal distance of 15 inches. Calculate his BCF and BCS for lifting a tire and export your results using the export button.
3. Pick some task that involved you lifting an object in the past. Determine the BCF and BCS for the task in mind and export your results!

The scenarios were designed to slowly get the participant more comfortable with the Backster™ web application. The first scenario gave the user all the information along with step-by-step instructions. The second scenario forced the user to think about how far a tire gets held

from a person. Occupational safety officers performing safety evaluations on a manufacturing floor had to guess this value; they were not able to accurately measure a worker's lift without disturbing the worker. Thus, in industry, a user might not have all the inputs explicitly given to them. The final scenario made the survey more personal and showcased how Backster™ fit into a person's life.

Finally, a study participant answered questions on the post-survey. The first question on the survey gave seventeen statements and had the participants say if they strongly disagreed, disagreed, neither agreed nor disagreed, agreed, or strongly agreed with the statements. This portion of the survey was used to gather how participants liked the overall web application, font, graphics, color scheme, and tool tips. The last few statements gauged how the participant viewed the usefulness of the tool in their lives and if the participant trusted Backster™ to prevent the user from attempting a dangerous lift. The team collected how often participants planned on using Backster™. The last three questions were open ended. Two questions collected three of the most positive and negative aspects of the Backster™ web application. The final question gave the participant a voice in stating what one change they would make to Backster™ if they had the ability to do so. The last question of the survey helped the participants feel as though their responses would have a change in the web application.

5.2 Results before BCF and BCS Scales Implemented

With the controlled study, all participants were graduate students. Seventy-five percent of the students mentioned they had knowledge or training related to user interface design. Only twenty-five percent of the participants stated they had knowledge or training on occupational safety and ergonomics. All contributors stated they used Google Chrome as their Internet browser.

The post survey results showed how the Backster web application satisfied majority of users. They found the application easy to use and were comfortable with the web application. The participants enjoyed the interface's design and font. All of these graduate students felt confident in their ability to use Backster™. However, only half of the participants agree that it was easy to find help or hints if an item was not clear. Three other statements scored low, meaning that fifty percent or less either agreed or strongly agreed with the statements. The following lists the statements:

- Backster™'s calculations make sense to me.
- I see Backster™ as being a useful tool in my daily life.
- I trust Backster™ to tell me if I am about to attempt a dangerous/harmful lift.

With majority of these participants not having a background in occupational safety or ergonomics, it made sense they would not know how Back Compressive Force or Back Compressive Stress were calculated. Developers needed to add a page describing how the formulas work and how to interpret the numerical results.

Since these graduate students were not studying occupational safety, the Backster™ team did not expect the students to say they would use Backster™ regularly.

The lack of transparency for the calculations played a major role in participants stating they did not trust Backster™ to keep them from performing a harmful lift.

As for the open-ended feedback, the top three positive aspects were enjoyment of the graphics, the color scheme, and the ease of use. Half of the participants did not understand the results.

The responses for the question about significantly changing one element of the web application varied. Since the scenarios for the questionnaire were written before the product was

fully developed, the scenarios gave a person's height in inches whereas the application wanted the height in feet and inches. One participant did not like having to perform the mental calculation. Another participant suggested making the skeleton grow and shrink as the height value changes. A third response concerned the necessity for an explanation of the results.

Based on the results of the controlled study, the team concluded an about Backster™ page was necessary for people unfamiliar with occupational safety and ergonomics to understand how the tool worked. Otherwise, the developed features satisfied these users.

5.3 Results after BCF and BCS Scales Implemented

With the field study, all participants were graduate students with previous exposure to occupation safety, ergonomics, and user interface design. The graduate students were enrolled in the Occupational Safety Engineering (OSE) Master's Program. The students had also taken a human factors course that introduced them to the concept of user interface design. These participants had four browsers that scored equally in terms of use: Internet Explorer, Safari, Firefox, and Google Chrome.

The post survey results showed how the Backster web application satisfied majority of users in the field study. They, too, found the application easy to use and were comfortable with the web application. The participants enjoyed the interface's design, the posture graphics, and the font choice. They also felt confident in their ability to use the Backster™ web application. While majority of the results favored the web application, six statements scored low, meaning that fifty percent or less either agreed or strongly agreed with the statements. The following lists the low scoring statements:

- It was easy to find help/hints if an item was not clear.
- The user interface was pleasant.

- I like the design of the web application.
- I am satisfied with using Backster™ for calculations.
- I see Backster™ being a useful tool in my daily life.
- I trust Backster™ to tell me if I am about to attempt a dangerous/harmful lift.

In order to reveal the hints, a user had to hover the cursor over a particular element. This proved to not be the most efficient manner of helping a user. This problem needed to be addressed with a new how-to webpage or tutorial page for those unfamiliar with the Backster™ web application.

The open feedback questions did not enlighten the team as to why these OSE students found the interface unpleasant or dislike the design. The group decided to view the problem as a disagreement on aesthetics and not worry about finding a solution at that time.

The team found it very interesting that OSE students did not trust the Backster™ calculations and were not satisfied with the calculations given their knowledge on the subject. However, the open-ended questions revealed an insight into the OSE students' perspectives.

The OSE students commented they believed users would benefit more from having the acronyms of BCF and BCS written out. They also mentioned that BCF did not tell the whole story when it came to determine if a lift was safe for an individual. One noted that frequency and duration needed to be utilized in a calculation of cumulative back compressive force, but the first iteration of Backster™ was only concerned with calculating the BCF and BCS results for a one-time lift.

While some had minor issues with Backster™, one declared this web application was a “great teaching instrument to demonstrate the effect of posture”. The near instantaneous results and ease of use were other positive aspects to Backster™.

The results from the field study closely mirrored those of the controlled study even though the students had different background. The web application satisfied most participants with an OSE background while meeting the client's requirements.

5.4 Pre-Survey Results

The pre-survey significant questions are:

1. Do you have any knowledge or training pertaining to Occupational Safety or Ergonomics?

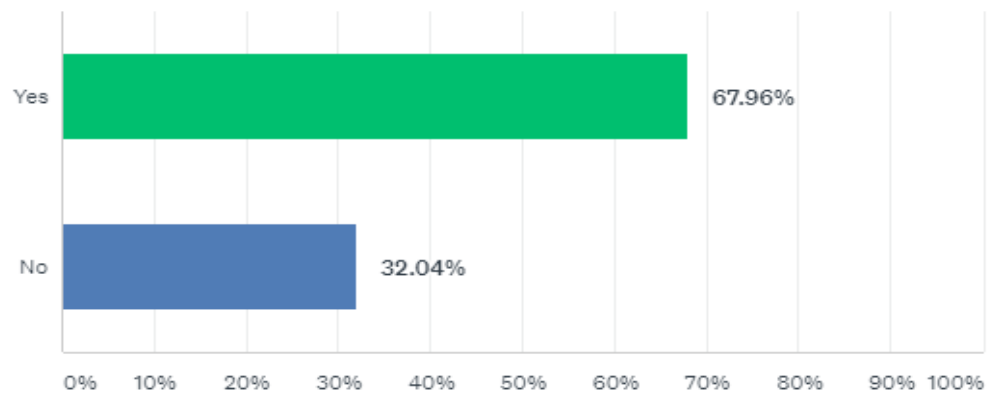


Figure 22: Survey result for background

Most of the users are students in the Industrial and Systems Engineering department. A few of the users had industry experience up to 10 years in Occupational safety or Ergonomics.

2. Do you have any knowledge or training pertaining to User Interface Design (UID)?

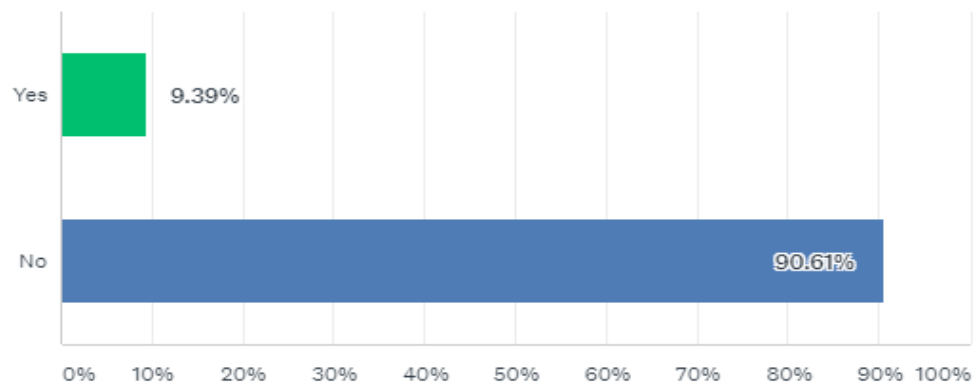


Figure 23: Survey result for knowledge on UID

3. Are you currently a student or are you working in industry?

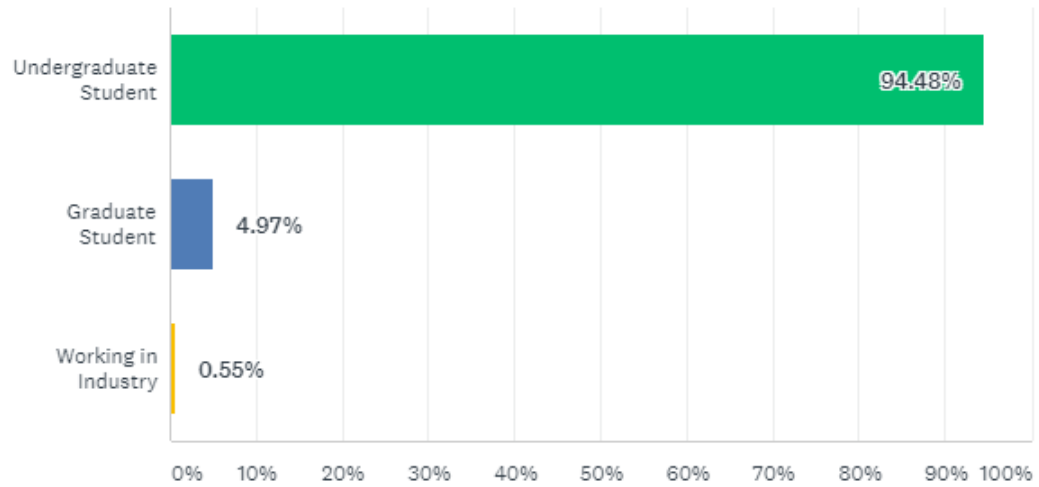


Figure 24: Survey result about Demographics

4. What internet browser do you use mostly? Select all that apply.

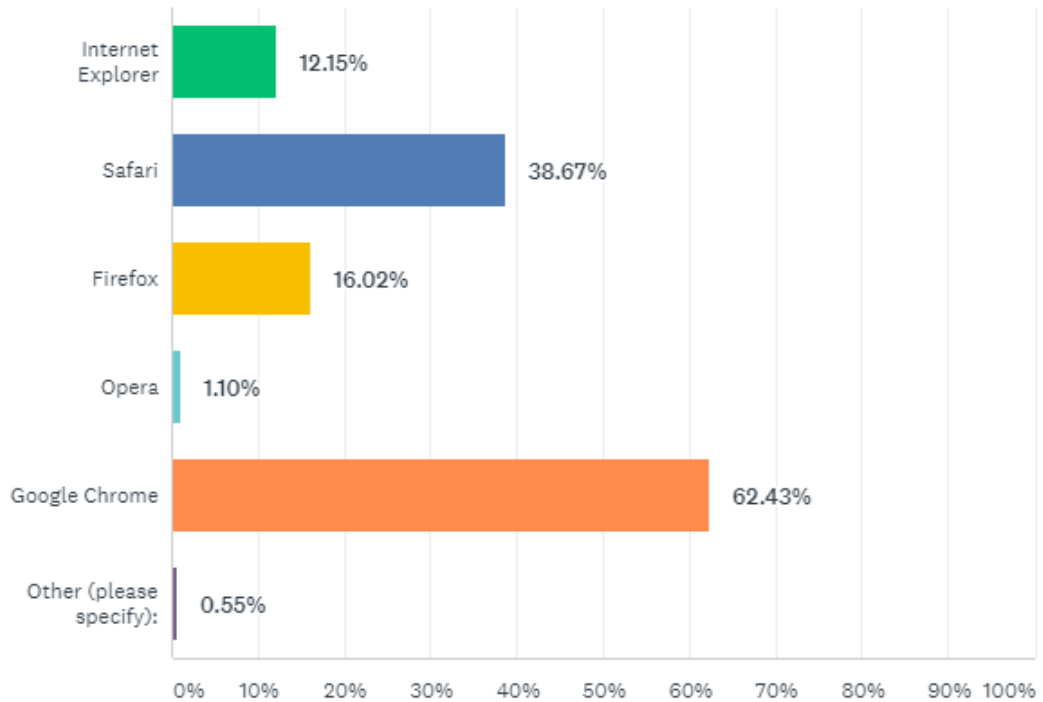


Figure 25: Survey result about browsers

5.5 Post-Survey Results

The post-survey questions related to **Ease of Use** are:

1. Backster™ is easy to use.

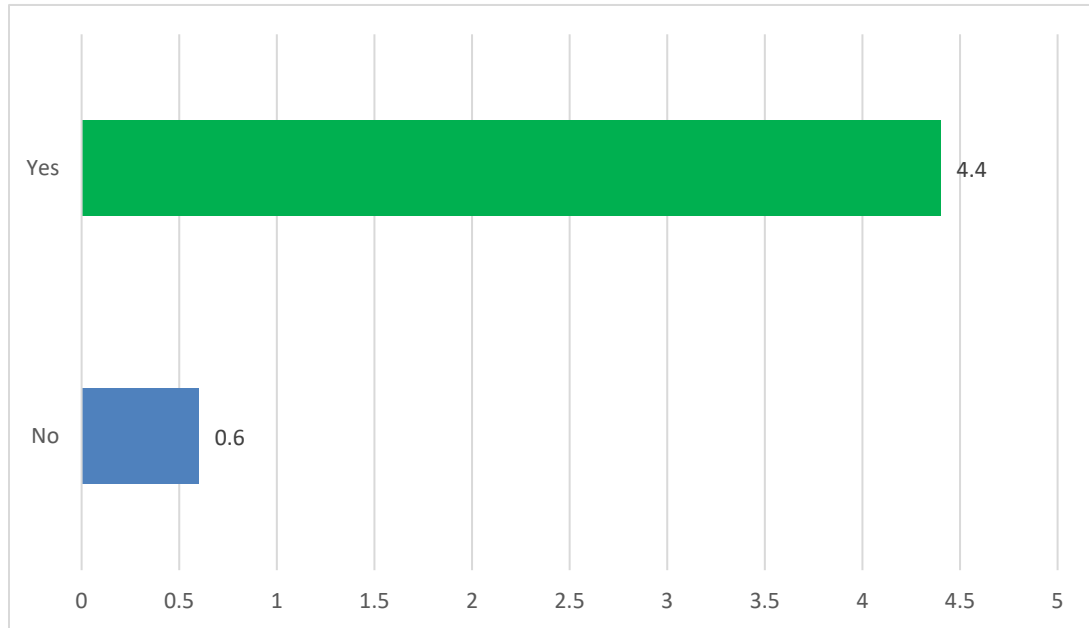


Figure 26: Backster is easy to use

2. I feel comfortable using the Backster™ web application.

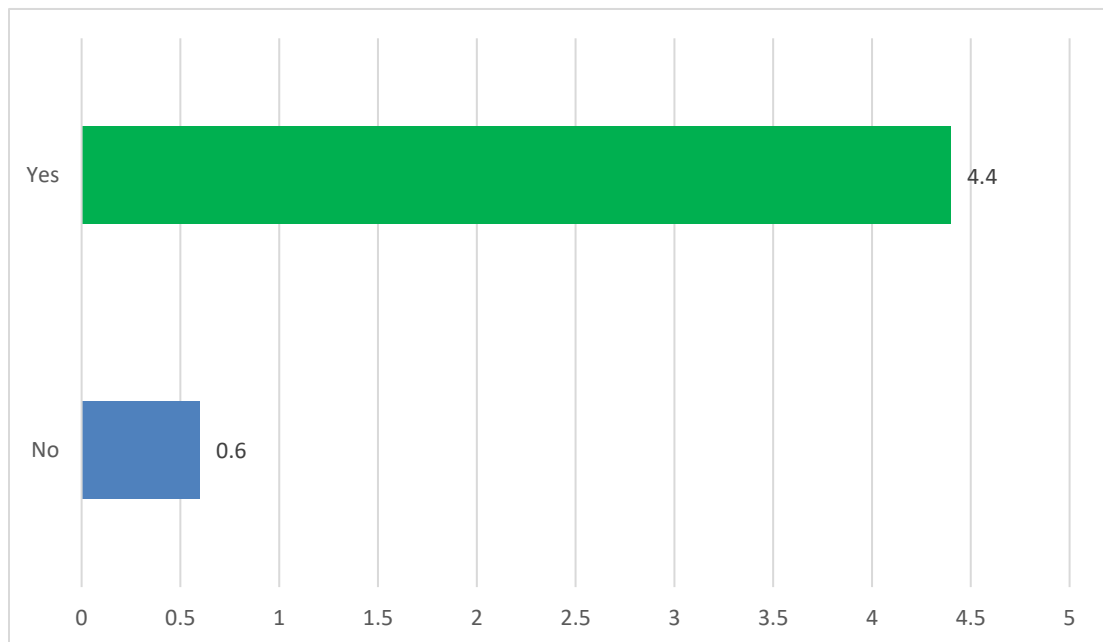


Figure 27: Comfortable using the Backster application

3. It was easy to find help/hints if an item was not clear.

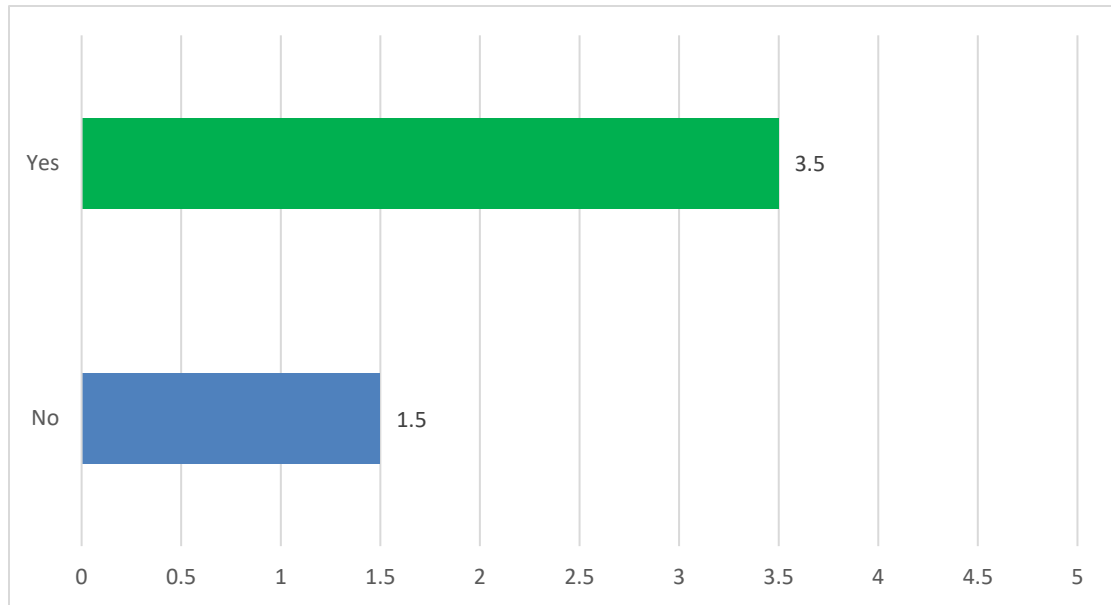


Figure 28: Easy to find help/hints

The post-survey questions related to **Effective Feedback** are:

1. I found there to be an appropriate balance between graphics and text for feedback.

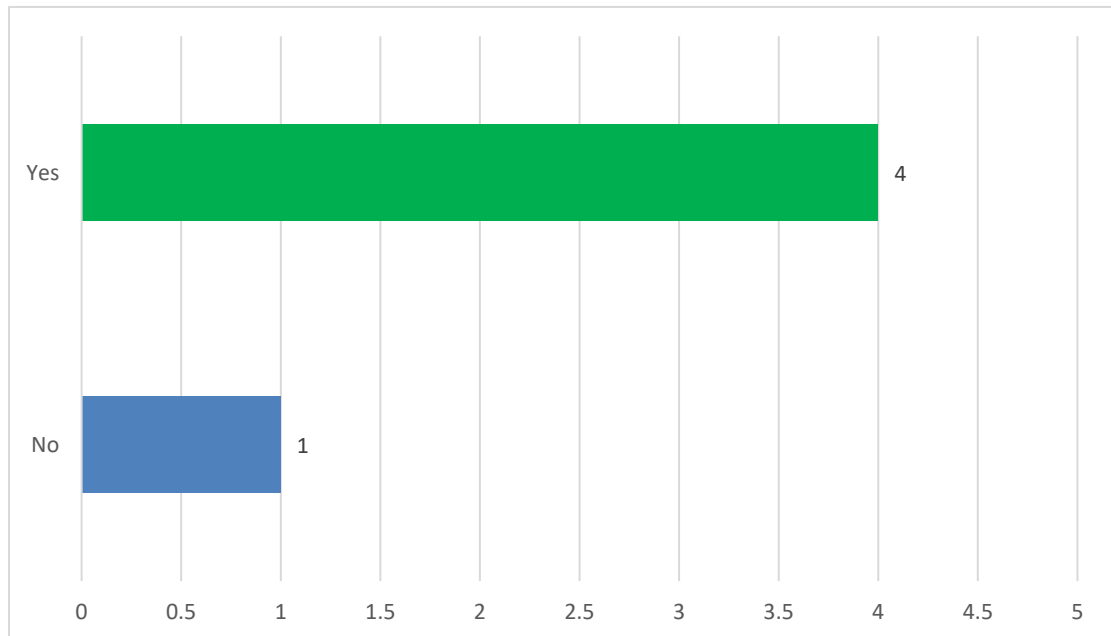


Figure 29: Balance between graphics and text for feedback

2. Backster™'s calculations make sense to me.

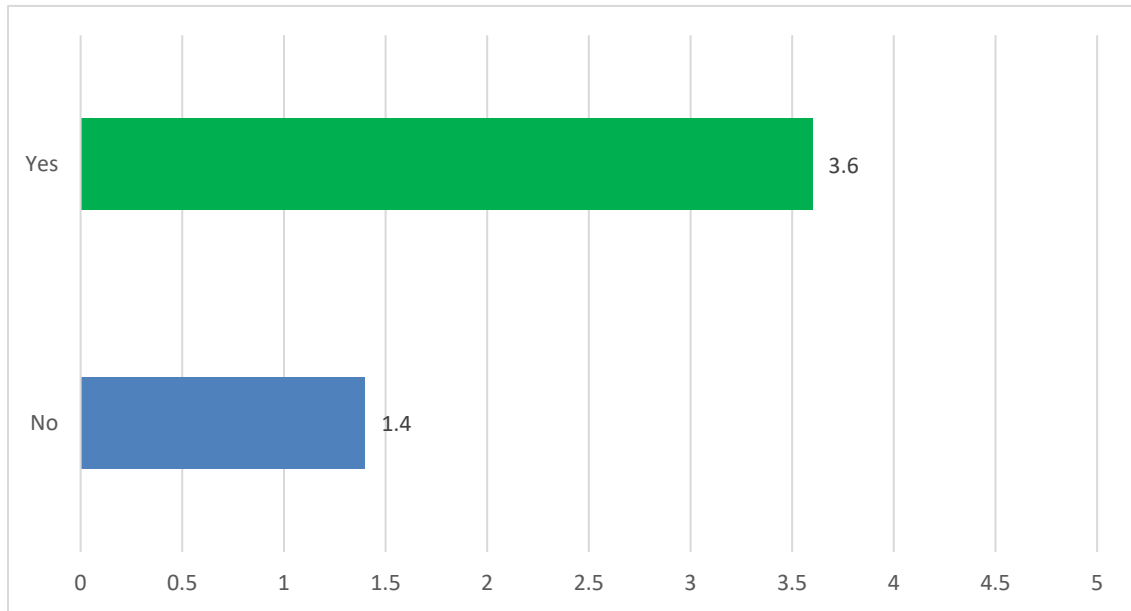


Figure 30: Backster Calculation make sense

3. I trust Backster™ to tell me if I am about to attempt a dangerous/harmful lift.

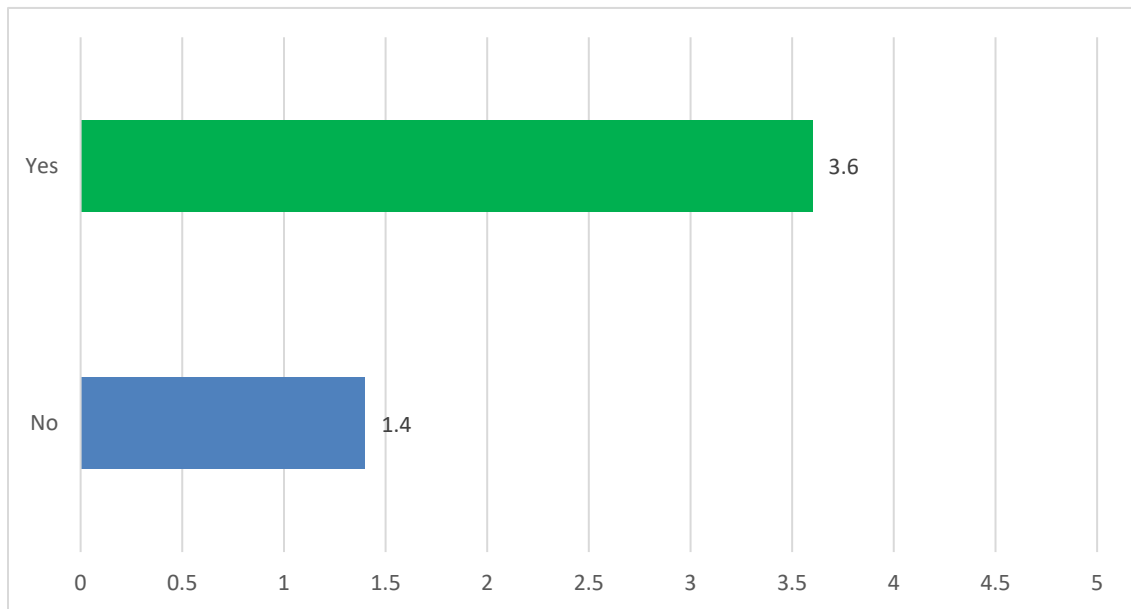


Figure 31: Trust Backster

The post-survey questions related to **Aesthetically Pleasing** are:

- 1. The user interface was pleasant.

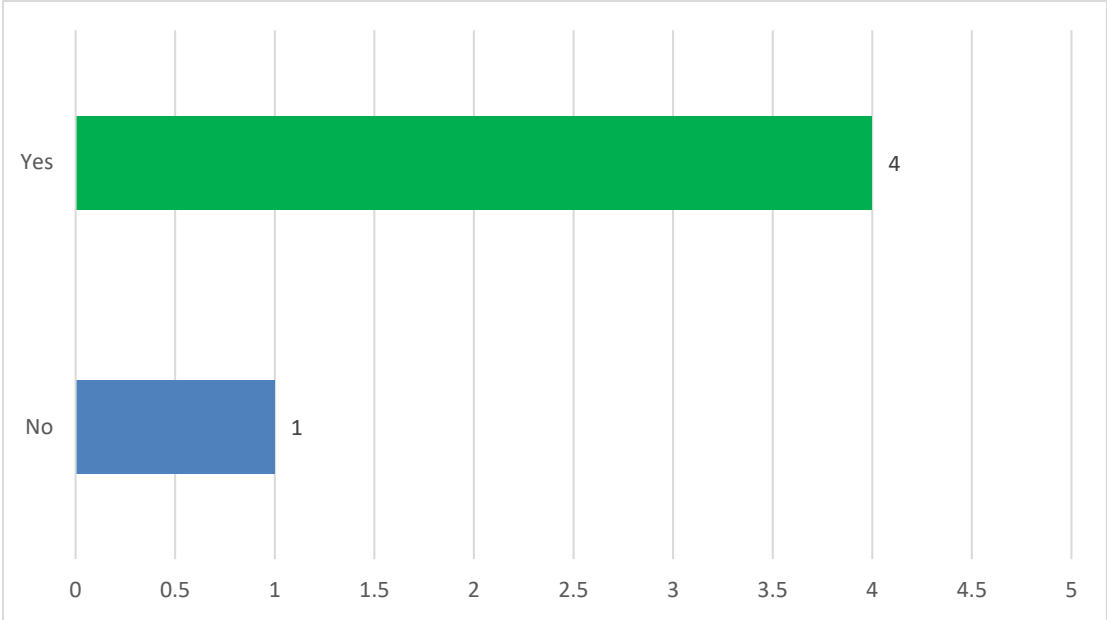


Figure 32: The user interface was pleasant

- 2. I like the design of the web application.

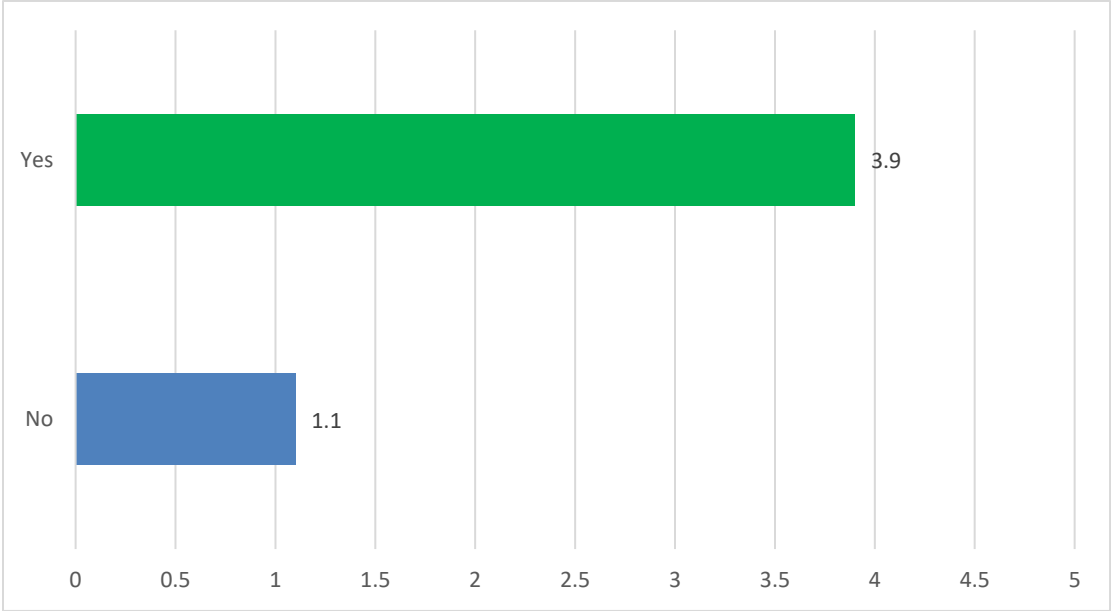


Figure 33: Like the design of the application

3. The graphics used in Backster™ are pleasing.

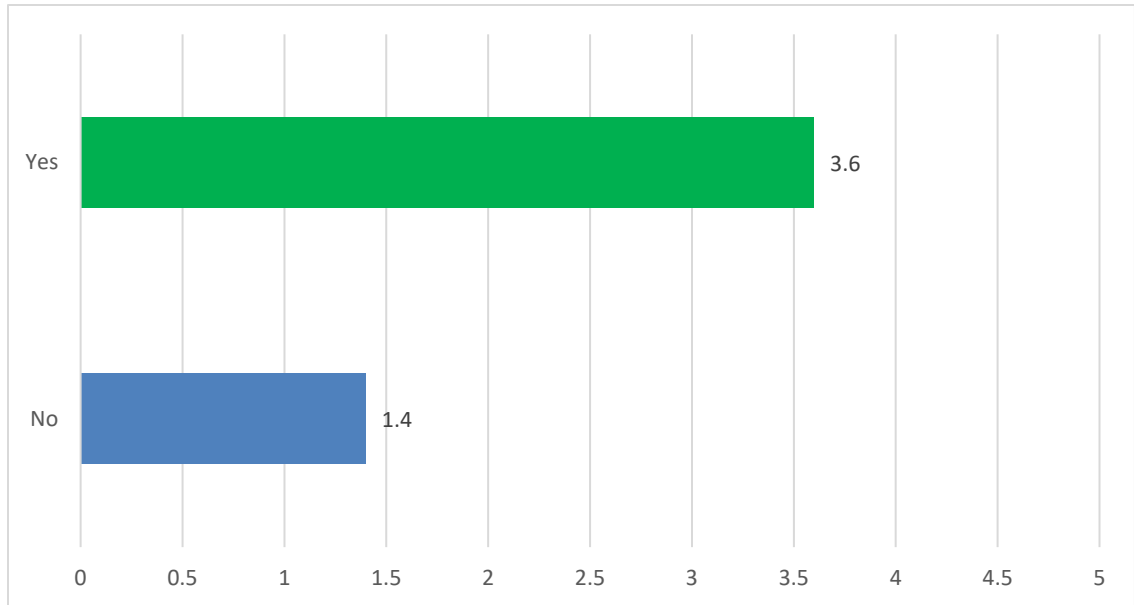


Figure 34: Graphics are Pleasing

The post-survey questions related to **Satisfaction** are:

1. The color scheme used with Backster™ is attractive.

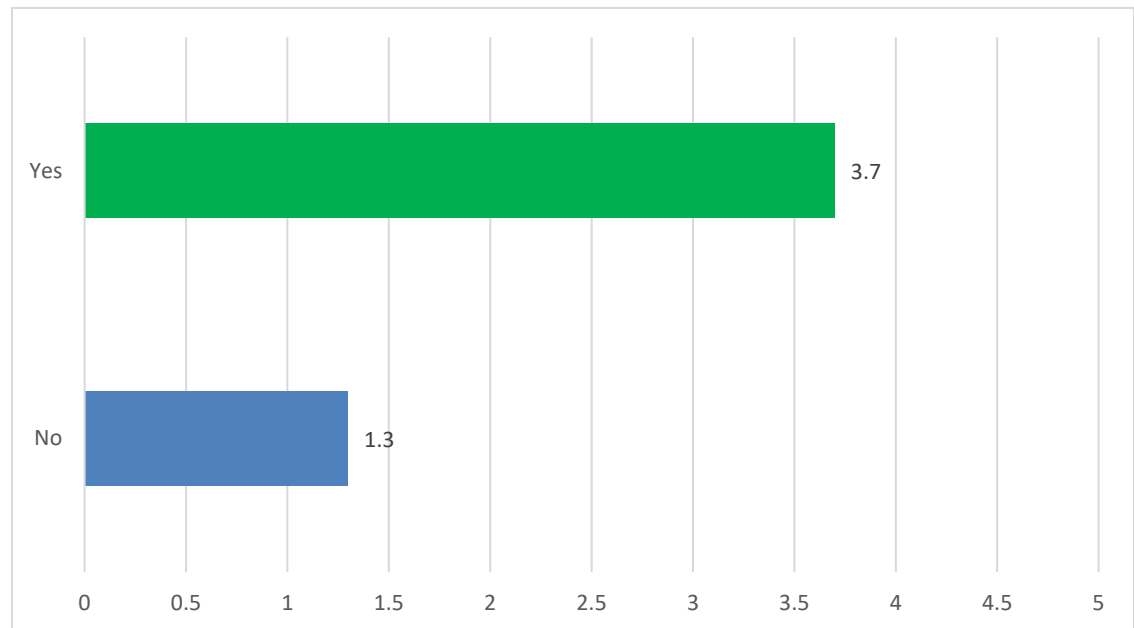


Figure 35: Color scheme is attractive

2. I understand the information Backster™ is asking me for.

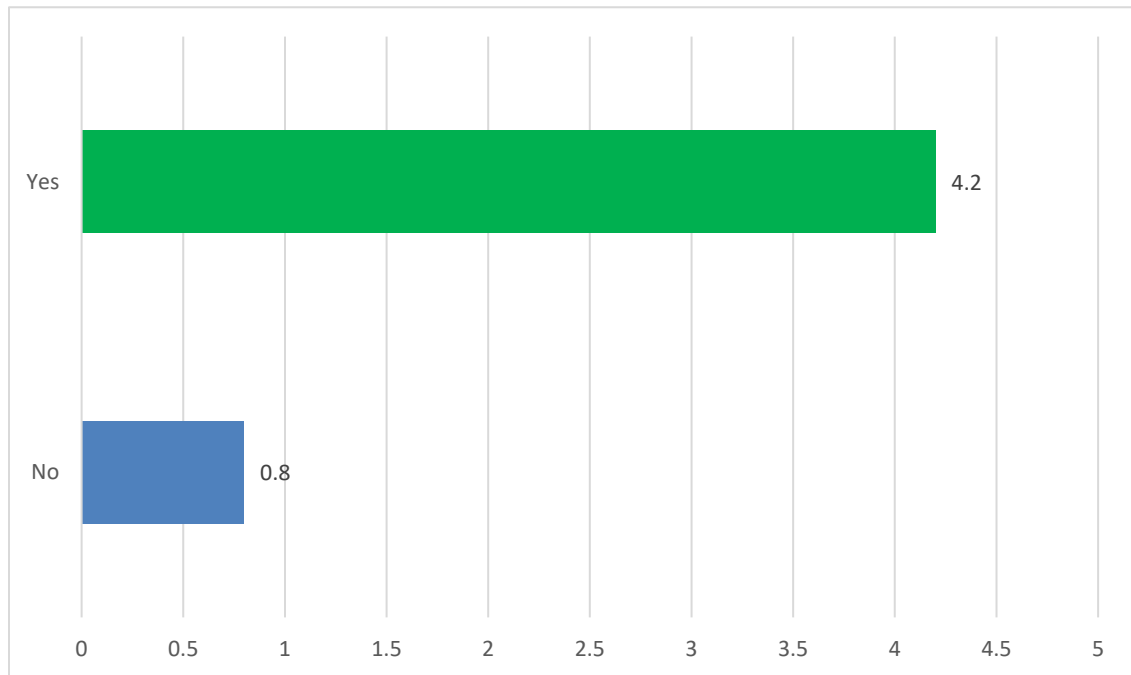


Figure 36: Understand the information application is asking me

3. The typography/fonts used with Backster are easy to read.

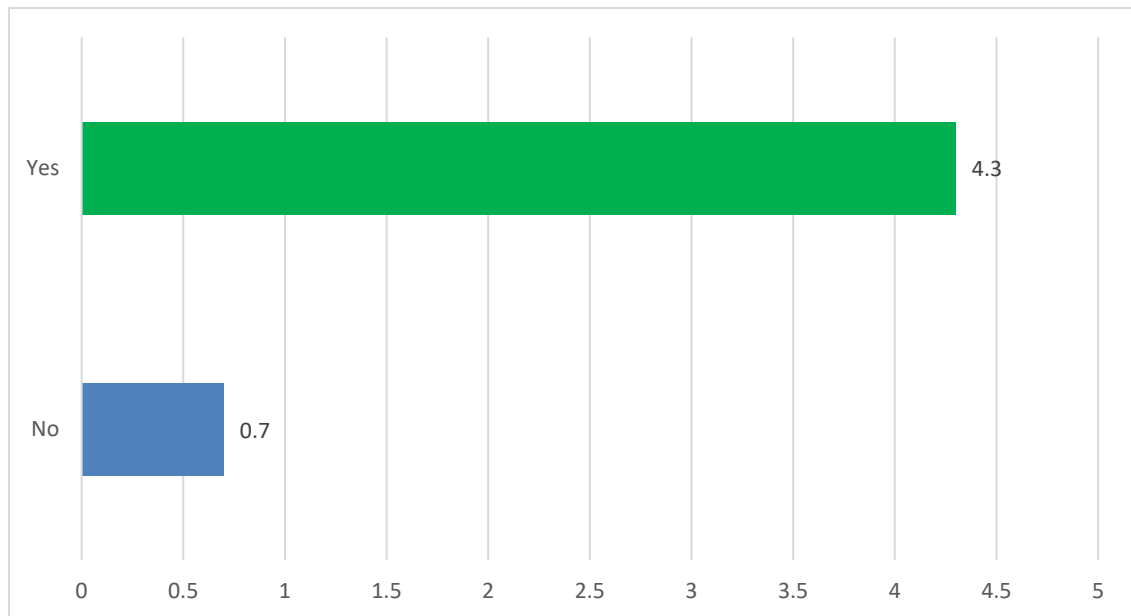


Figure 37: Typography/fonts are easy to read

The post-survey questions related to **Feature Provided** are:

1. I am satisfied with using Backster™ for calculations

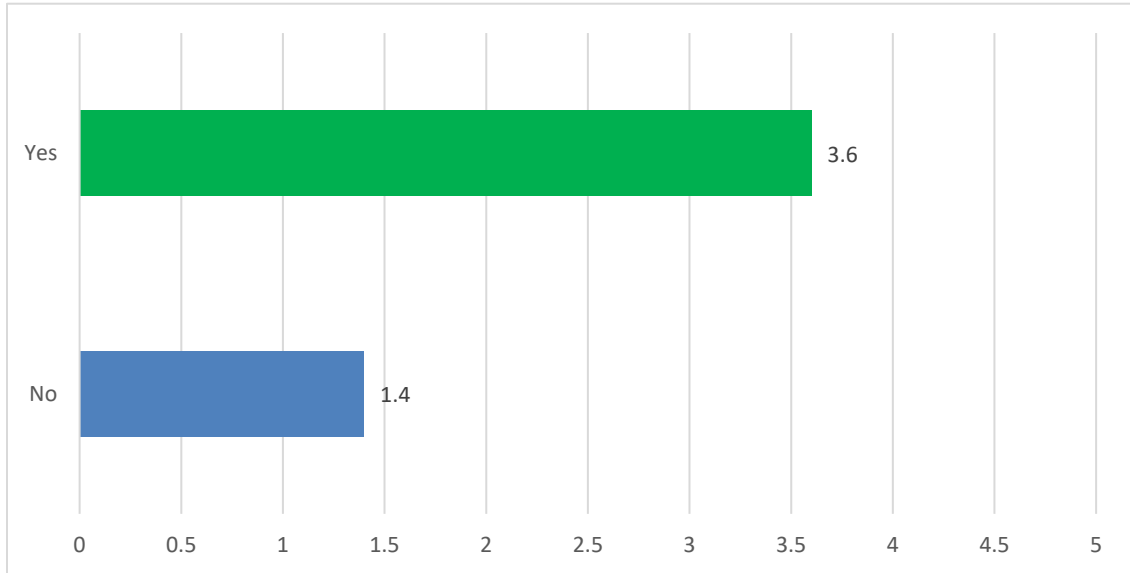


Figure 38: Satisfied with calculations

2. I found the application very intuitive

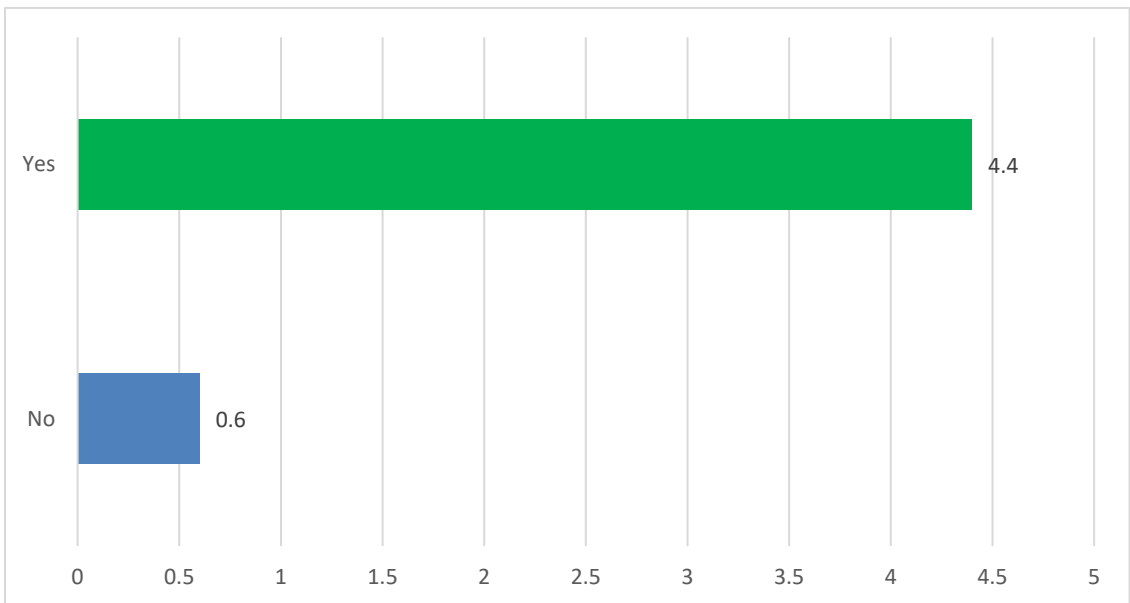


Figure 39: Application Intuitive

3. I have confidence in my ability to use the Backster™ web application.

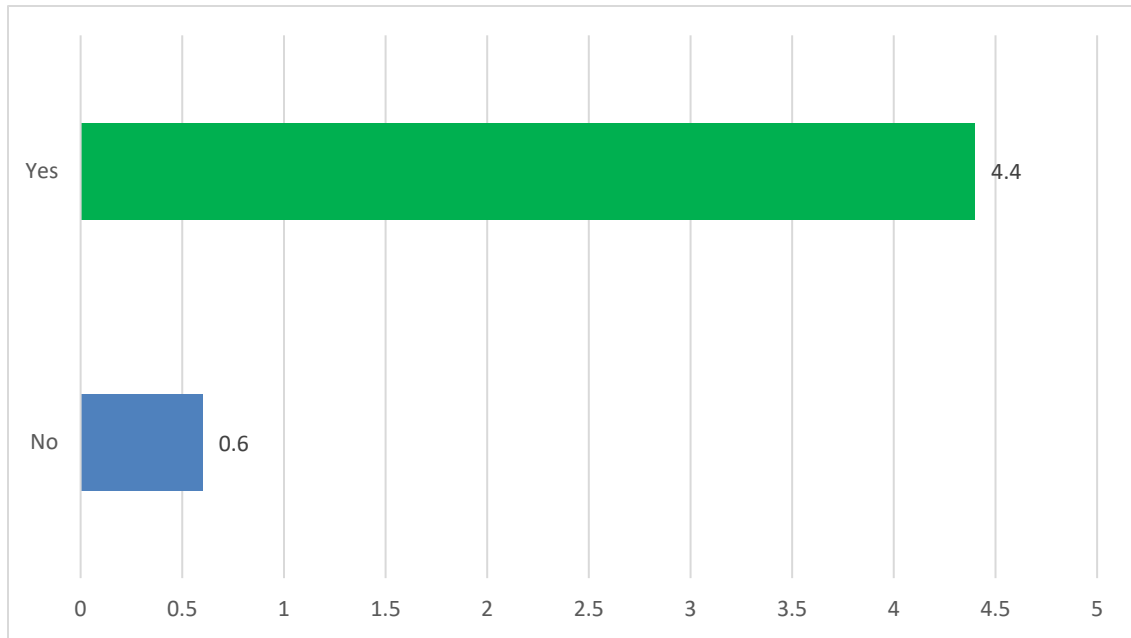


Figure 40: Confidence in ability to use Backster

5.6 Analysis of All Responses

Overall, both groups of participants agreed on Backster™ being easy to use. The posture graphics were a highlight for many using the web application. The design also had a desirable balance between graphics and text. The text was also revealed to be easy to read and pleasant. Majority of the participants also enjoyed the color scheme of the web application.

The study revealed some major issues with the current Backster™ prototype. Users wanted the calculations to be more transparent. The lack of transparency caused a significant amount of distrust with respect to the calculations. It was a necessity that users, whether they be students learning about ergonomics or occupational safety managers, trust the calculations to be correct and kept individuals from performing dangerous lifts. A potential solution to this problem involved adding two pages to the web application that explain how the Back Compressive Force and Back Compressive Stress were calculated and a tutorial on how to properly use the web application.

5.7 Advantages of using MeteorJS

MeteorJS allows one to create Layouts, Templates and render them when ever needed. The application contains two Layouts, one for the Ergonomist dashboard and one for all other pages. The Layouts contain header and footer information and we use them across all pages other than Ergonomist dashboard. By using Layouts and templates, we reduced duplicate code (boilerplate code) by code reuse. MeteorJS provides several packages, such as accounts-password for managing the user accounts, iron-router for routing, etc. MeteorJS can generate mobile app both Android and iOS. To generate a mobile app, a mobile-config.js is required to specify the app name, author information, email, server information, icons, loading screen, and any other preferences. To generate the Android app, we require: JDK 1.8 or higher, Android SDK, Gradle tool for build automation, a private key to sign the app (we used Keytool by Oracle Java), and Sing the app with the private key (we used jarsigner by Oracle Java). A meteor build command is used to generate

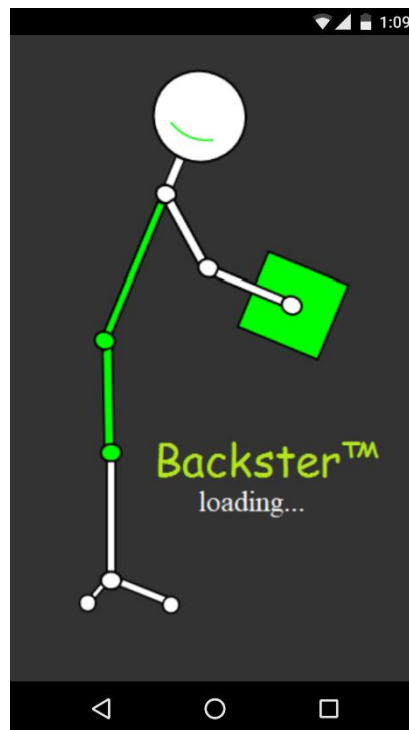


Figure 41: Backster™ Android app loading screen

the mobile app. Figure 41 shows the loading page and Figure 42 shows the home page of Backster™ Android mobile app.

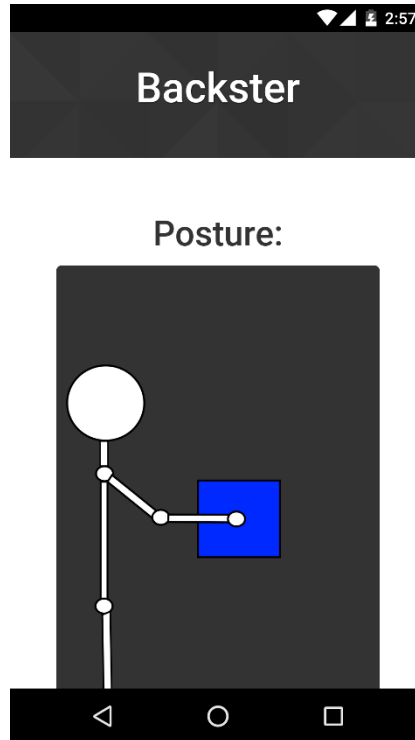


Figure 42: Backster™ Android app homepage

5.8 Performance Improvements

Initial color coding is done on JPEG images, which require at least 40 images. Resizing the posture is not possible with JPEG images. The JPEG images cannot adapt to different screen sizes. The size of the 40 JPEG images was 7.95 MB whereas the size of SVG is just 2.75 KB. So, by using SVG the efficiency of the application is improved.

For managing user data, the load, move, clear data methods are created. The methods performance is analyzed and improved by using an alternate approach. The move data from CurrentData collection to AllData collection is implemented using two approaches. First one is with BulkOp of MongoDB and the second one is removing the data instead of moving. Figure 43 shows the time (milliseconds) taken for the load, move, and clear data methods. Figure 43 shows

the time (milliseconds) taken by the clear data method for different number of documents in the MongoDB collection. Figure 44 shows the time (milliseconds) taken to insert user data into one collection vs inserting into two collections.

```
I20171115-00:35:23.145(-6)? clearData: took 268.168545 milliseconds for 588 records
=> Meteor server restarted
I20171115-00:38:16.074(-6)? clearData: took 106.479829 milliseconds for 1 records
I20171115-00:38:44.954(-6)? clearData: took 108.83278 milliseconds for 10 records
I20171115-00:39:00.527(-6)? clearData: took 153.259503 milliseconds for 20 records
I20171115-00:39:22.322(-6)? clearData: took 86.987777 milliseconds for 74 records
```

Figure 43: Performance measure of clear data method

```
=> App running at: http://localhost:3000/
Type Control-C twice to stop.

I20171114-05:10:41.608(-6)? saveData: took 111.6623 milliseconds for both inserts
I20171114-05:11:08.147(-6)? saveData: took 87.099333 milliseconds for both inserts
=> Meteor server restarted
I20171114-05:12:12.497(-6)? saveData: took 59.657316 milliseconds for one insert
I20171114-05:12:18.352(-6)? saveData: took 45.342826 milliseconds for one insert
I20171114-05:12:35.756(-6)? saveData: took 43.727934 milliseconds for one insert
I20171114-05:12:46.589(-6)? saveData: took 44.697137 milliseconds for one insert
=> Meteor server restarted
I20171114-05:13:19.892(-6)? saveData: took 112.307989 milliseconds for both inserts
I20171114-05:13:29.664(-6)? saveData: took 90.898503 milliseconds for both inserts
I20171114-05:13:37.798(-6)? saveData: took 88.999365 milliseconds for both inserts
```

Figure 44: Performance measure of save data method

```
=> App running at: http://localhost:3000/
Type Control-C twice to stop.

I20171115-00:57:55.299(-6)? loadData: took 22391.857533 milliseconds for 500 records
I20171115-00:57:59.165(-6)? loadData: Final took 26259.138712 milliseconds for 588 records
I20171115-00:58:41.865(-6)? moveData: took 24341.003367 milliseconds for 500 records
I20171115-00:58:45.876(-6)? moveData: Final took 28352.491906 milliseconds for 588 records
I20171115-00:59:19.840(-6)? loadData: took 24892.793715 milliseconds for 500 records
I20171115-00:59:23.704(-6)? loadData: Final took 28756.25966 milliseconds for 588 records
I20171115-00:59:29.788(-6)? clearData: took 138.651844 milliseconds for 588 records
```

Figure 45: Performance measure of move data and load data methods

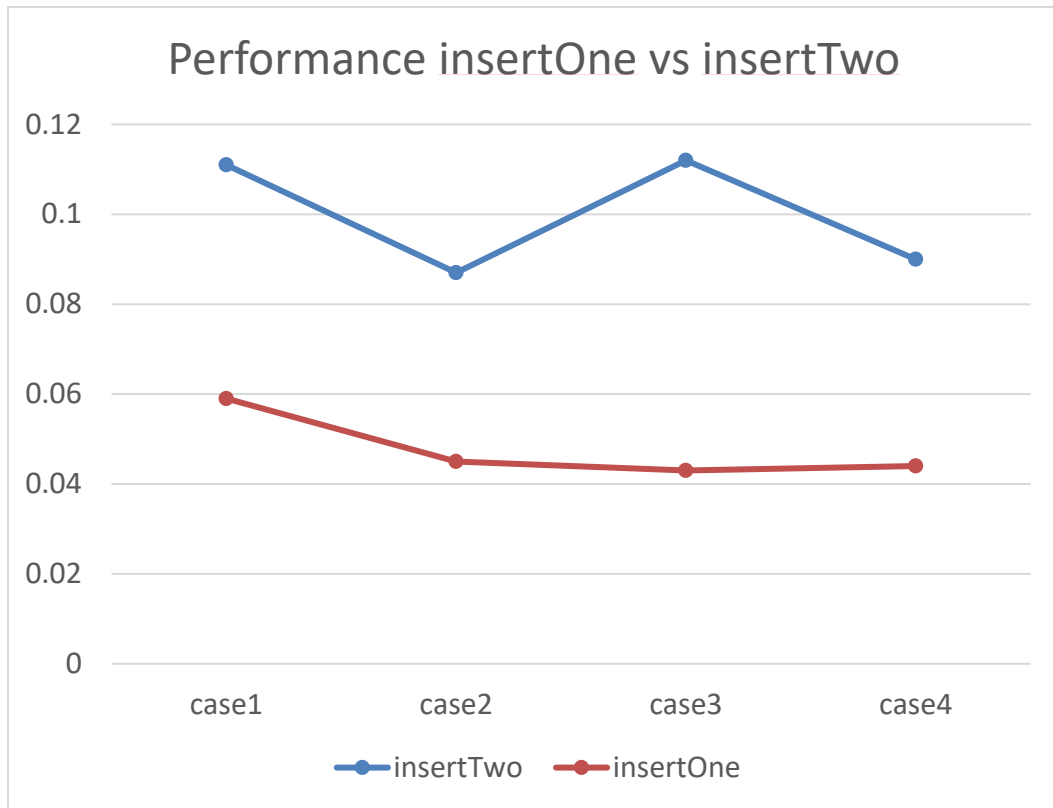


Figure 46: Performance comparison between one and two inserts

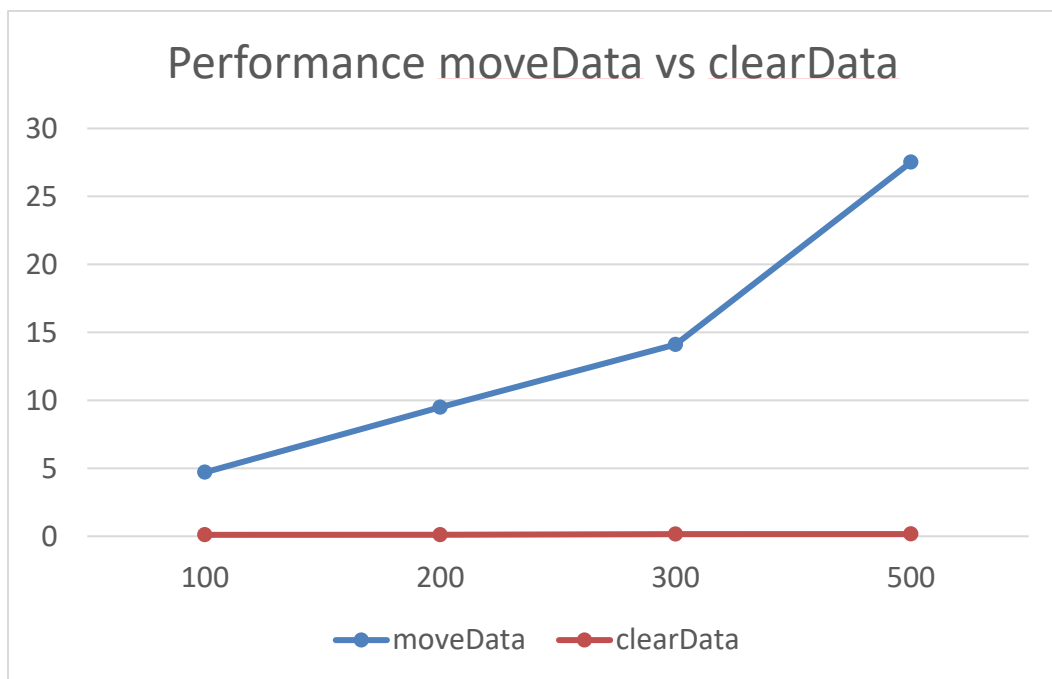


Figure 47: Performance comparison between moveData vs clearData

Figure 47 shows the performance measure comparison between inserting into one collection and two collections. Figure 46 shows performance measure comparison between move

data and clear data methods. The load data method is also inefficient as it takes more time, so implemented a mechanism that switches between the AllData and CurrentData collections based on the load data and clear data options are clicked. The getStats method calculates stats from AllData collection if load option is selected and CurrentData collection if clear option is selected. Figure 48 shows the time (milliseconds) taken by the load data method with new mechanism of switching between the collections.

```
=> App running at: http://localhost:3000/  
Type Control-C twice to stop.  
I20171117-01:02:13.714(-6)? loadData: Final took 86.501391 milliseconds for 879 records
```

Figure 48: Performance measure of load data method

It was assured that the end-product is easy to use and consistently yields reliable results by utilizing the best technologies available.

CHAPTER 6: CONCLUSION AND FUTURE WORK

The Backster™ application utilizes values specified by a user and produced the Back Compressive Force and Back Compressive Stress estimates for the individual attempting a particular lift. By the end of the research, the Backster™ web application provided the user with the feedback on a particular lift. The web application provides a better Ease of use, feedback mechanism and better overall experience to the Ergonomist/user. The look and feel of web application was designed in such a way that any user can comfortably utilize the features offered. The questionnaire results showed user found the interface design appealing. The customer reported that he was highly satisfied with the product at the end of the research.

The application allows the users to calculate the result for a one-time lift. The Ergonomist/Client can draw charts with the user data. For future work, the integration of repetitive task result calculation is a good next step. This supports the users to calculate the result for both one-time lift and repetitive task. The other ergonomic tools can be integrated, to make this an Ergonomist toolbox. The repetitive task can be added to both the user calculation and the dashboard, so that the charts can be drawn for both one-time and repetitive tasks. The dashboard supports moving and loading of data, but all the data is combined once moved/loaded. To improve the efficiency of the system, creating an Ergonomist/Client data management dashboard, which supports the move/load each session data to/from a collection in the database. The application supports login and registration, future work includes password change/recovery and email verification. Although the application can adapt to all screen sizes, but creating native iOS and Android application provides a better user experience for the mobile users.

REFERENCES

- [1] Occupational safety and health. (2017, March 3). In Wikipedia, The Free Encyclopedia. Retrieved March 24, 2017, from https://en.wikipedia.org/w/index.php?title=Occupational_safety_and_health&oldid=808611286
- [2] The Importance of Safety. Retrieved March 26, 2017, from <http://www.worksafemt.com/safety/safety-important/the-importance-of-safety/>
- [3] Loertscher, M., Merryweather, A. and Blowski, D. 2009. A revised back compressive force estimation model for ergonomic evaluation of lifting tasks. (2009). DOI = <http://dx.doi.org/10.3233/WOR-2009-0924>.
- [4] Shoubi, M., Barough, A. and Rasoulijavaheri, A. 2013. Ergonomics Principles and Utilizing it as a Remedy for Probable Work Related Injuries in Construction Projects. *International Journal of Advances in Engineering & Technology*. 6, 1 (2013), 232-245.
- [5] Thiese, M., Hegmann, K., Wood, E., Garg, A., Moore, J., Kapellusch, J., Foster, J. and Ott, U. 2014. Prevalence of low back pain by anatomic location and intensity in an occupational population. *BMC Musculoskeletal Disorders*. 15, 1 (2014), 283.
- [6] Miñón, R., Moreno, L., Martínez, P. and Abascal, J. 2014. An approach to the integration of accessibility requirements into a user interface development method. *Science of Computer Programming*. 86, (2014), 58-73. DOI = <http://dx.doi.org/10.1016/j.scico.2013.04.005>.
- [7] Galitz, W. 2007. *The Essential Guide to User Interface Design: An Introduction to GUI Design*. Wiley Pub.
- [8] Baturay, M. and Birtane, M. 2013. Responsive Web Design: A New Type of Design for Web-based Instructional Content. *Procedia - Social and Behavioral Sciences*. 106, (2013), 2275-2279. DOI = <http://dx.doi.org/10.1016/j.sbspro.2013.12.259>.
- [9] Voutilainen, J., Salonen, J. and Mikkonen, T. 2015. On the Design of a Responsive User Interface for a Multi-device Web Service. *2015 2nd ACM International Conference on Mobile Software Engineering and Systems*. (2015). DOI = <http://dx.doi.org/10.1109/MobileSoft.2015.16>.
- [10] D. Li and L. Deng, "Multi-resolution Mechanism for SVG," 2009 Asia-Pacific Conference on Information Processing, Shenzhen, 2009, pp. 139-143. DOI: 10.1109/APCIP.2009.171 URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5197156&isnumber=5197113>
- [11] Nielsen, J. 1994. Usability inspection methods. Conference companion on Human factors in computing systems - CHI '94. (1994). DOI = <http://dx.doi.org/10.1145/259963.260531>.
- [12] Ghazarian, A. *How to Conduct a Usability Heuristic Evaluation - Designmodo*. Retrieved 11 November 2017, from <http://designmodo.com/usability-heuristic-evaluation/>
- [13] Kujala, S., Roto, V., Väänänen-Vainio-Mattila, K., Karapanos, E. and Sinnelä, A. 2011. UX Curve: A method for evaluating long-term user experience. *Interacting with Computers*. 23, 5 (2011), 473-483. DOI = <http://dx.doi.org/10.1016/j.intcom.2011.06.005>.
- [14] *Back Injuries Fact Sheet. UMD -Department of Environmental Safety, Sustainability & Risk*. Retrieved 10 April 2017, from <https://www.essr.umd.edu/back-injuries-fact-sheet>

- [15] *Back Safety*. OSU -Department of Environmental Health and Safety. Retrieved 12 April 2017, from <https://ehs.okstate.edu/content/back-safety-1>
- [16] *Nonfatal Occupational Injuries and Illnesses Requiring Days Away From Work, 2015*. (2015). U.S. Bureau of Labor Statistics. Retrieved 12 April 2017, from <https://www.bls.gov/news.release/osh2.nr0.htm>.
- [17] *OSHA Technical Manual (OTM) | Back Disorders and Injuries | Occupational Safety and Health Administration*. U.S. Department of Labor. Retrieved 12 April 2017, from https://www.osha.gov/dts/osta/otm/otm_vii/otm_vii_1.
- [18] *NLE Calc: A Mobile Application Based on the Revised NIOSH Lifting Equation*. U.S. Centers for Disease Control and Prevention. Retrieved 16 April 2017, from <https://blogs.cdc.gov/niosh-science-blog/2017/08/25/nleapp/>
- [19] *Auburn University: Occupational Safety, Ergonomics, and Injury Prevention*. Auburn University -Samuel Ginn College of Engineering. Retrieved 16 April 2017, from <http://eng.auburn.edu/research/centers/occupational-safety-ergonomics-injury-prevention/index.html>
- [20] *Lifting calculator app*. Oregon Department of Consumer and Business Services. Retrieved 16 April 2017, from <http://osha.oregon.gov/OSHAPubs/apps/liftcalc/lifting-calculator-app.html>
- [21] Scrum (software development). (2017, June 6). In *Wikipedia, The Free Encyclopedia*. Retrieved June 25, 2017, from [https://en.wikipedia.org/w/index.php?title=Scrum_\(software_development\)&oldid=809007440](https://en.wikipedia.org/w/index.php?title=Scrum_(software_development)&oldid=809007440)