

6-13-2017

# Track My Hoist

Brett Harrison

Santa Clara University, [b1harrison@scu.edu](mailto:b1harrison@scu.edu)

Madison Rubia

Santa Clara University, [mrubia@scu.edu](mailto:mrubia@scu.edu)

Paras Zaveri

Santa Clara University, [pzaveri@scu.edu](mailto:pzaveri@scu.edu)

Follow this and additional works at: [https://scholarcommons.scu.edu/cseng\\_senior](https://scholarcommons.scu.edu/cseng_senior)



Part of the [Computer Engineering Commons](#)

---

## Recommended Citation

Harrison, Brett; Rubia, Madison; and Zaveri, Paras, "Track My Hoist" (2017). *Computer Engineering Senior Theses*. 95.

[https://scholarcommons.scu.edu/cseng\\_senior/95](https://scholarcommons.scu.edu/cseng_senior/95)

This Thesis is brought to you for free and open access by the Engineering Senior Theses at Scholar Commons. It has been accepted for inclusion in Computer Engineering Senior Theses by an authorized administrator of Scholar Commons. For more information, please contact [rscroggin@scu.edu](mailto:rscroggin@scu.edu).

**SANTA CLARA UNIVERSITY  
DEPARTMENT of COMPUTER ENGINEERING**

Date: June 7, 2017

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY  
SUPERVISION BY

**Brett Harrison, Madison Rubia, and Paras Zaveri**

ENTITLED

**TRACK MY HOIST**

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING**



THESIS ADVISOR



DEPARTMENT CHAIR

**SANTA CLARA UNIVERSITY**  
**DEPARTMENT of COMPUTER ENGINEERING**

---

**TRACK MY HOIST**

---

**June 13, 2017**

Brett Harrison, Madison Rubia, Paras Zaveri

**Advisor:** Ronald Danielson, PhD

# **TRACK MY HOIST**

by

Brett Harrison, Madison Rubia, and Paras Zaveri

## **SENIOR DESIGN PROJECT REPORT**

Submitted in partial fulfillment of the requirements  
for the degree of  
Bachelor of Science in Computer Science and Engineering  
School of Engineering  
Santa Clara University

Santa Clara, California

June 13, 2017

**SANTA CLARA UNIVERSITY**  
**DEPARTMENT of COMPUTER ENGINEERING**

Date: June 13, 2017

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY  
SUPERVISION BY

**Brett Harrison, Madison Rubia, and Paras Zaveri**

ENTITLED

**TRACK MY HOIST**

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE  
DEGREE OF

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING**

---

THESIS ADVISOR

---

DEPARTMENT CHAIR

# Contents

1 List of Figures . . . . .	3
2 Abstract . . . . .	4
3 Introduction . . . . .	5
4 Requirements . . . . .	7
4.1 Functional Requirements . . . . .	7
4.2 Non-Functional Requirements . . . . .	7
4.3 Design Constraints . . . . .	7
5 Use Case Diagram . . . . .	9
6 Use Cases . . . . .	10
7 Activity Diagram . . . . .	13
8 Conceptual Model . . . . .	15
9 Architectural Diagram . . . . .	24
10 Technologies Used . . . . .	25
11 Design Rationale . . . . .	27
11.1 Justification of User Experience . . . . .	27
11.2 Justification of Technologies Used . . . . .	27
12 Test Plan . . . . .	29
13 Risk Analysis . . . . .	31
14 Development Timeline . . . . .	33
15 Societal Issues . . . . .	35
15.1 Ethical . . . . .	35
15.2 Societal . . . . .	35
15.3 Political . . . . .	35
15.4 Economic . . . . .	35
15.5 Health and Safety . . . . .	36
15.5 Manufacturability . . . . .	36
15.6 Sustainability . . . . .	36

15.7 Environmental Impact . . . . .	36
15.8 Usability . . . . .	36
15.9 Lifelong learning . . . . .	37
15.10 Compassion . . . . .	37
16 Conclusion . . . . .	38
Appendix . . . . .	39
Project Files . . . . .	39

# 1 List of Figures

1	Use Case Diagram . . . . .	9
2	Admin Activity Diagram . . . . .	13
3	Employee Activity Diagram . . . . .	14
4	Index Page . . . . .	15
5	Register Page . . . . .	16
6	Login Page . . . . .	17
7	Home Page . . . . .	18
8	Register Lift Page . . . . .	19
9	Request Lift Page . . . . .	20
10	Employee Queue Details Page . . . . .	21
11	Admin Queue Details Page . . . . .	22
12	Admin Edit Queue Page . . . . .	23
13	Architectural Diagram . . . . .	24
14	Risk Analysis Table . . . . .	32
15	Gantt Chart . . . . .	34



## **2 Abstract**

Construction sites all over the world rely on hoist lifts for transporting material and workers at the job site. Despite being such an integral part of construction projects, there is currently no tool to track the use of this critical and highly necessary piece of equipment at the construction site. The purpose of this project is to design a system that would enable the managers and employees of construction companies to efficiently track and manage the hoist lifts at their sites.

Efficient tracking and management of the lifts would result in savings of cost, time, and energy for the construction companies. Such improvements would also help boost worker morale, which would consequently boost productivity.

Over the course of this project the team has designed an interactive yet intuitive software application that helps effectively track and manage hoist lifts on construction sites.

### 3 Introduction

Hoist lifts are an integral part of construction sites erecting high-rise buildings. Much of the work at the site is reliant on the availability of the hoist lift to transport construction material and contractors between various building levels. Despite advances in the construction industry, the hoist lift has not been upgraded to be more efficient for the companies that rent and use it on-site. Hoist lifts move at an average speed of ten floors per minute; it would take at least five minutes for the hoist lift to travel from the ground level to the top floor of a fifty-story building. Priority for the lift is biased towards lower floors, where travel times are shorter. This leads to workers on higher floors being unable to complete their assigned tasks for the day, as they are forced to wait indefinitely for hoist lift availability. The excruciatingly slow speed of these hoist lifts requires careful time management to avoid unnecessarily long wait times. For construction companies, the greatest consequence of inactivity is increased cost. Workers that are idle lose efficiency, leading to increased job hours, and longer jobs require a larger budget. Local construction sites report economic losses caused by the inefficiency of hoist lifts. With such glaring costs, a solution to efficiently manage and track the hoist lift would help the construction companies save time and money.

Currently, managers and supervisors at the job site do their best to keep the lift operating at all times. Radios are the primary form of on-site communication between workers and supervisors. The worker who is assigned to operate the lift relies on verbal calls, without any information about priority of current tasks. As of now, the only way to assess the current location of the lift is by visually estimating its position relative to the structure; however, this becomes a challenge on structures higher than ten stories. The current method for managing the hoist lift is a disorganized and informal queue that leads to indefinite wait times, frustrated workers, delayed productivity, and detrimental costs to construction companies.

Our solution is to provide construction teams with a desktop and mobile system that allows the workers to track the status of the hoist lift. The system will greatly benefit the construction companies by providing information that addresses the shortcomings of the current system for managing the hoist lifts. This solution will serve the workers on the site, as well as the supervisors who oversee the progress of construction. With the app, users can view in real time what floor the hoist lift is currently situated at, eliminating the wait time faced when relying solely on radio communication. This system benefits the supervisors by giving them greater ability to schedule and manage the lift activity, with access to the queue of tasks and requests currently waiting to be completed. Our system will not only indicate

the availability of the lift, but also notify when the lift has been idle for a set amount of time. Additionally, workers on the various floors will get real-time notifications shortly before their hoist lift is going to arrive based on position in the queue. With all this information, the site managers can analyze the data and implement *zoning\** on sites with more than one lift. All this data allows the supervisor to manage the hoist lift's current and subsequent tasks much more efficiently than before.

The display of real-time data pertaining to the accurate position of the hoist lift and approximate wait times will give the supervisors and the construction workers the power to manage their time effectively. With the information that our tool will provide, the prolonged and wasteful delays in productivity will be reduced, ultimately maximizing efficiency.

---

**\*zoning** - *A method implemented at construction sites with more than one hoist lift wherein the manager decides the floors of operation for a particular lift based on its usage. For instance, consider a thirty storey construction site with three hoist lifts. Assume that one lift is used more between the floors three to seven, the other between twelve to sixteen , and the third between nineteen to twenty four. Based on this data, the manager can implement the following zoning scheme:*

*Lift 1: Floors one to ten.*

*Lift2: Floors ten to nineteen.*

*Lift 3: Floors nineteen to thirty.*

*This is done in order to increase the efficiency related to the usage of the hoist lifts.*

## 4 Requirements

The requirements section defines and qualifies what our evaluation system needs to do. We communicated with the customer to gather the requirements.

### 4.1 Functional Requirements

Functional requirements define what the system must do.

The system will:

- Track the location of the hoist lift
- Provide separate admin and employee login
- Allow the users to send lift requests
- Allow the users to determine the task the lift is currently busy with and its status
- Provide a detailed report with stats like:
  1. Hoist idle time
  2. Direction of movement
  3. Amount of wait period between tasks and wait period of an employee while lift was busy with another task

### 4.2 Non-Functional Requirements

Non-functional requirements define the manner in which the functional requirements need to be achieved.

The system will be:

- User friendly and intuitive
- Reliable
- Responsive
- Real time

### **4.3 Design Constraints**

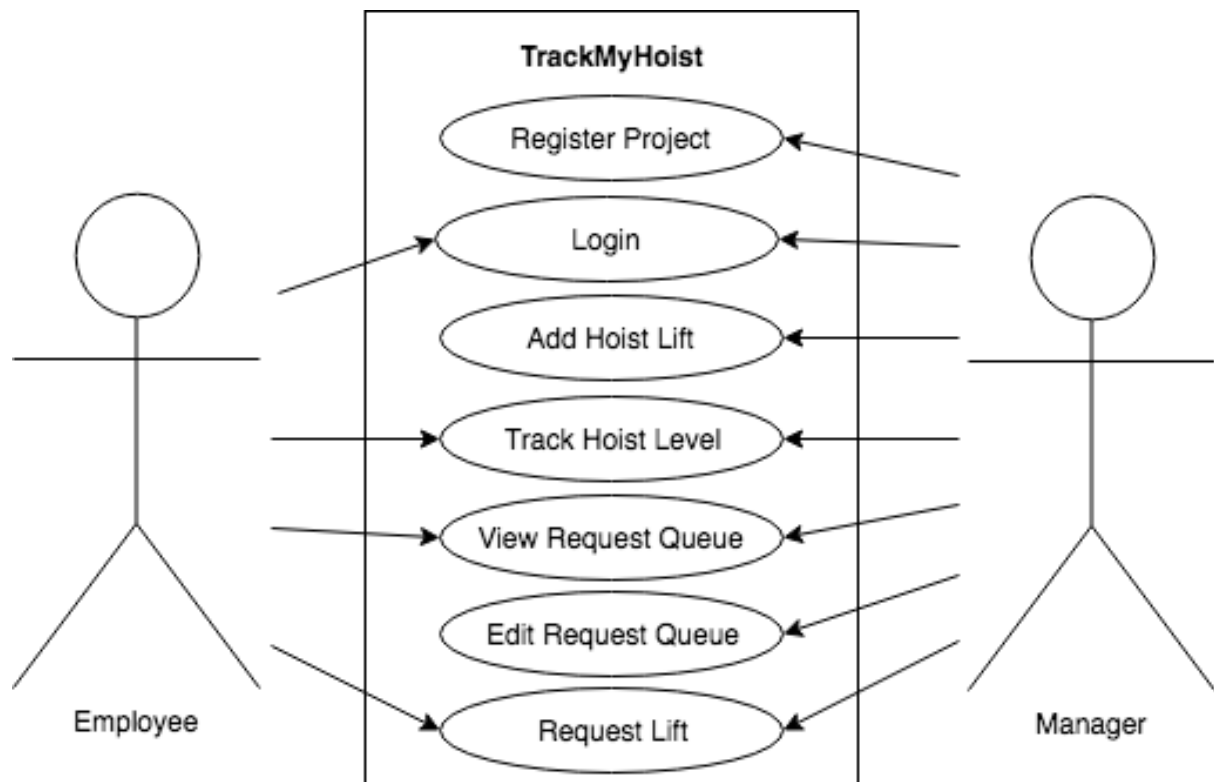
Design constraints are non-functional requirements that constrain the solution instead of the problem.

The system must:

- Be mobile and desktop friendly
- Run on various browsers and across all platforms and Operating Systems

## 5 Use Case Diagram

The use case diagram depicts the user's interactions with the system's various functionalities. Figure 1 displays the actors—the employee, and the manager—and the various actions they could take while using our system.



**Figure 1:** Use Case Diagram

## 6 Use Cases

Our system will have 7 use cases: register for the system, log in to the system, add hoist lifts to a project, track the current level of the hoist lift, view the queue of requests for the hoist lift, view and edit the request queue for the hoist lift, and request the hoist lift. For each use case, we describe the actors involved, the conditions of the use case, and the steps involved in executing the use case.

- Use Case 1: Registration Verification
  - Actors: Managers
  - Pre-Conditions:
    1. N/A
  - Steps:
    1. Open the app
    2. Click "Register" button
    3. Sign up with correct information
  - Post-conditions:
    1. See e-mail is received
- Use Case 2: Login Verification
  - Actors: Managers and Employees
  - Pre-Conditions:
    1. Registration Verification complete
  - Steps:
    1. Open the app
    2. Click "Login" button
    3. Log in with correct information
  - Post-conditions:
    1. If logged in with admin password, user is directed to admin page
    2. If logged in with employee password, user is directed to employee page
- Use Case 3: Add Hoist Lift Verification

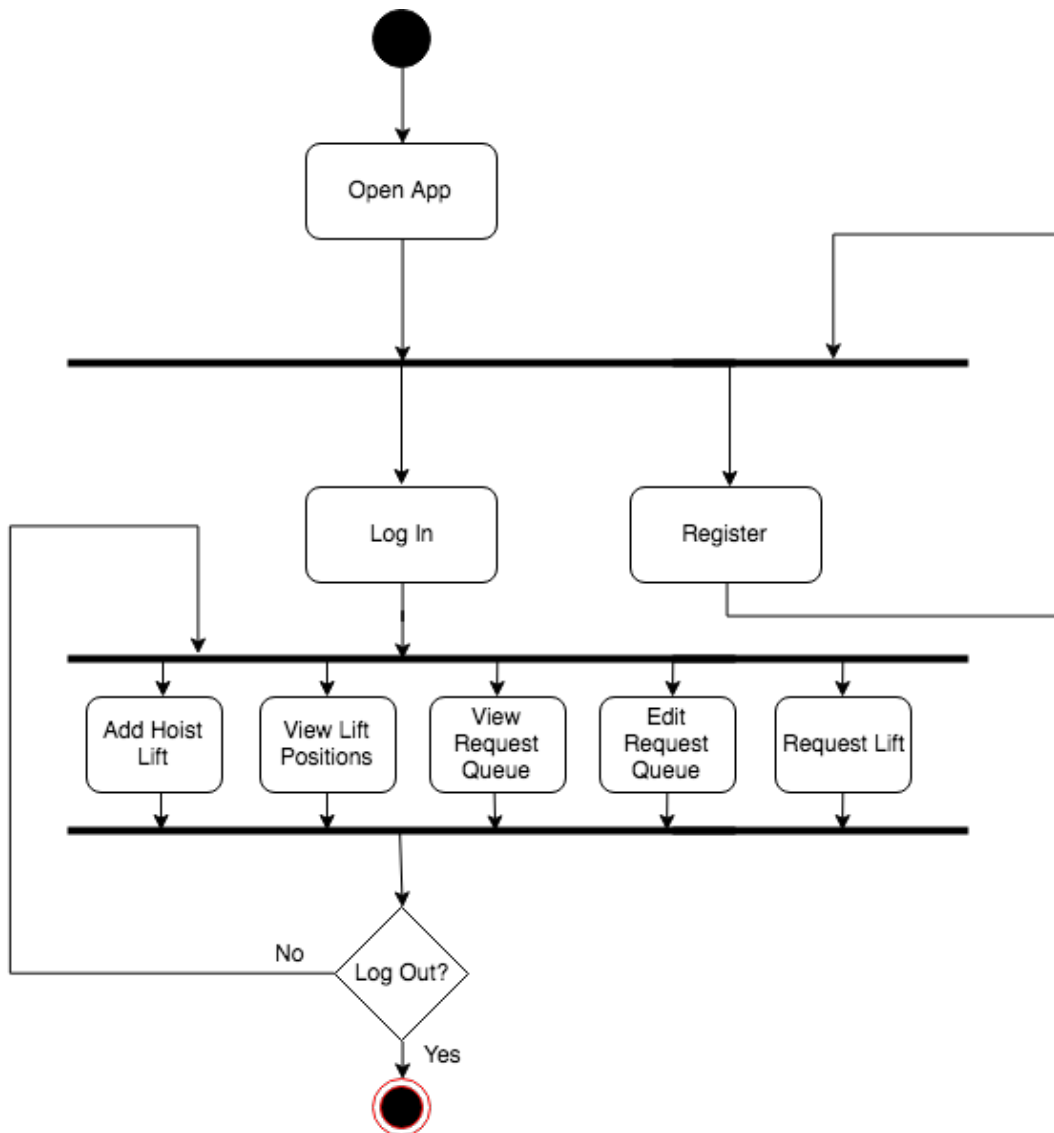
- Actors: Managers
- Pre-conditions:
  1. Login Verification complete
  2. Manager has project registered on the site
  3. Manager is logged in
- Steps:
  1. Click "Register Lift"
  2. Enter correct number of lifts to be added
  3. Click "Submit" button
- Post-conditions:
  1. See lift added to main page
- Use Case 4: Track Hoist Level
  - Actors: Managers and Employees
  - Pre-conditions:
    1. Add Hoist Lift Verification complete
  - Steps:
    1. N/A
  - Post-conditions:
    1. See display of correct on-site hoist lifts on home page
- Use Case 5: View Hoist Queue Verification
  - Actors: Managers and Employees
  - Pre-conditions:
    1. Add Hoist Lift Verification complete
  - Steps:
    1. Click "View Queue" button under desired hoist lift
  - Post-conditions:
    1. See table of requests for desired hoist lift



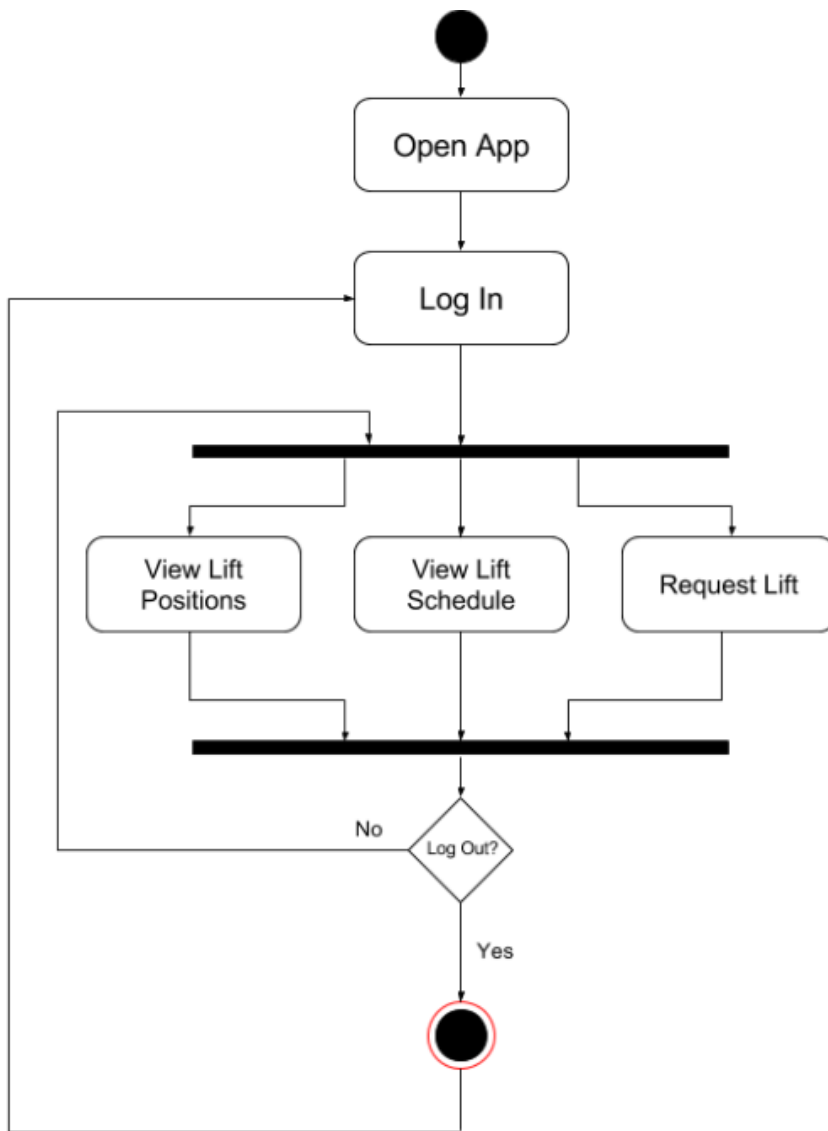
- Use Case 6: Edit Queue of Requests
  - Actors: Managers
  - Pre-conditions:
    1. View Hoist Queue complete
  - Steps:
    1. Select "View Queue" button under desired hoist lift
    2. Select "Edit Queue" button under request queue table
    3. Select "Remove" button next to desired items to delete
  - Post-conditions:
    1. See successfully updated queue table
  
- Use Case 7: Request Hoist Lift
  - Actors: Managers and Employees
  - Pre-conditions
    1. View Hoist Queue Verification complete
  - Steps:
    1. Click "Request" button
    2. Fill out request form with correct information
    3. Click submit
  - Post-conditions
    1. See request added to the queue for the correct lift

## 7 Activity Diagram

The activity diagram details the user's path while using the system. Figure 2 displays the paths an admin could take, while Figure 3 displays the paths an employee could take.



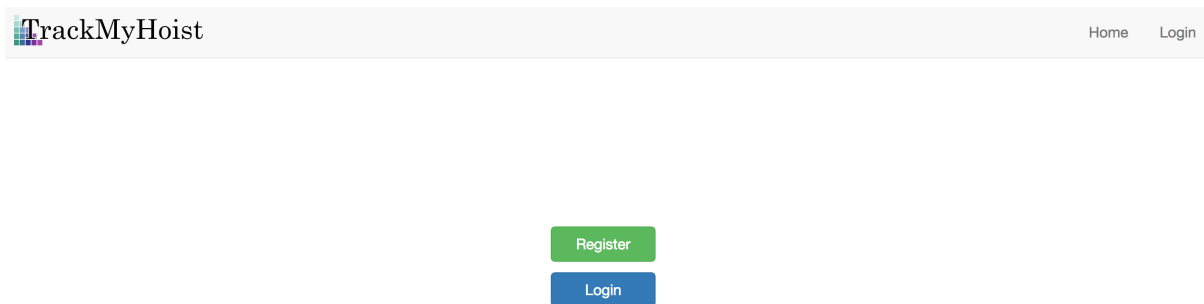
**Figure 2:** Admin Activity Diagram



**Figure 3:** Employee Activity Diagram

## 8 Conceptual Model

The conceptual model showcases the final product. Figure 4 shows our Index Page, to which the user will be routed when they enter the website URL.



**Figure 4:** Index Page

Figure 5 shows the Registration Page for new users. A new user is requested to provide the Project Name, the Company, their name and email, an admin password, and a general employee password.

TrackMyHoist Home Login

back

Register Project

Project Name

Company

Name

Email

Admin Password\*

Employee Password\*\*

\*Admin Password is the main password for admins to manage the project

\*\*Employee Password is the general password for all regular employees to login

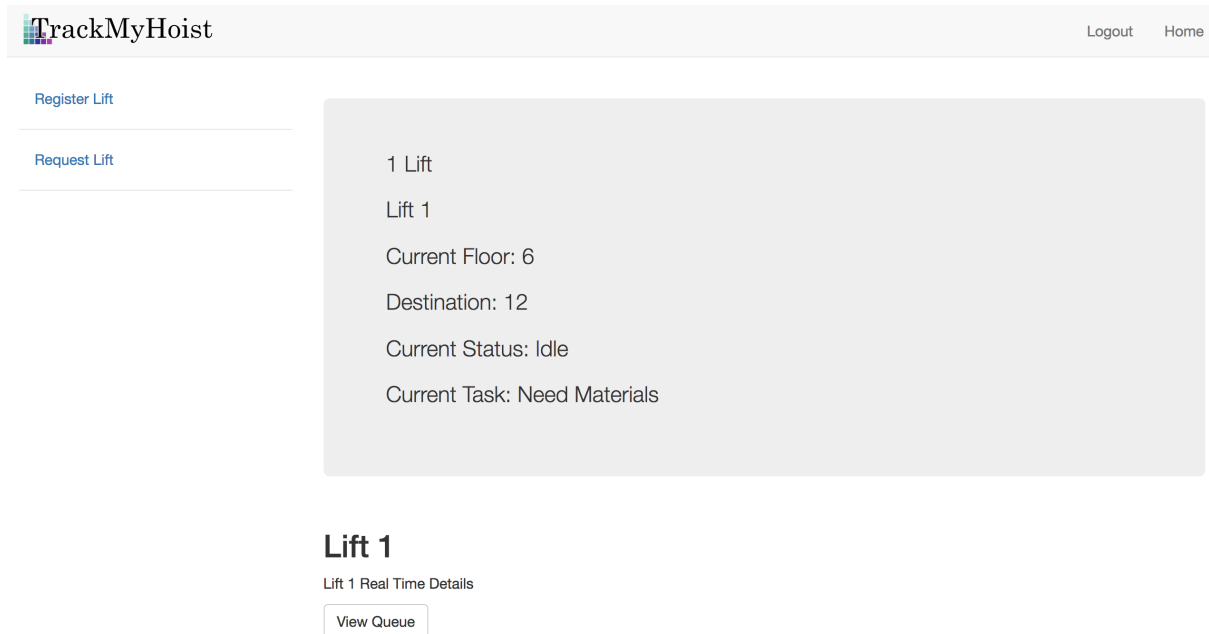
**Figure 5:** Register Page

Figure 6 shows the Login Page. Users must enter their email and password to access the site pages.

The screenshot displays the login interface for Track My Hoist. At the top left is the site logo, and at the top right are links for 'Home' and 'Login'. The login form is centered and includes a blue 'back' button at the top left. Below it is the 'Login' heading. The form contains two input fields: 'Email' and 'Password'. At the bottom of the form is a blue 'Log In' button.

**Figure 6:** Login Page

Figure 7 Shows the Home Page. The real-time info about each lift is displayed, including current floor, current task, and velocity of the lift.



**Figure 7:** Home Page

Figure 8 shows the Request Lift Page. Admins enter the number of lifts to be added and submit.

The screenshot shows the Track My Hoist application interface. At the top left is the logo 'TrackMyHoist' and at the top right are the links 'Logout' and 'Home'. The main content area is a white box containing a 'back' button at the top left. Below it is a horizontal line. Underneath the line is a text input field with the placeholder text 'Number of Lifts to Add'. At the bottom of the input area is a blue 'Submit' button.

**Figure 8:** Register Lift Page



Figure 9 shows the Request Lift Page. Here users can request the use of a specified lift to a specified floor, with the task description an optional field.

The screenshot shows the 'Request Lift Page' interface. At the top left is the 'TrackMyHoist' logo, and at the top right are 'Logout' and 'Home' links. The main content area contains a 'back' button, followed by a horizontal line. Below this are three input fields: 'Lift Number\*' (required), 'Floor\*' (required), and 'Task' (optional). A note '\*Required Fields' is positioned below the 'Task' field. At the bottom of the form is a blue 'Submit' button.

**Figure 9:** Request Lift Page

Figure 10 shows the Employee Queue Details Page. Employees logged into the site can view the current queue of tasks for a certain lift, including floor requested and task involved.

The screenshot shows the TrackMyHoist interface. At the top left is the logo and name 'TrackMyHoist'. At the top right are links for 'Logout' and 'Home'. Below the header is a 'back' button. The main content area is titled 'Lift Queue' and contains a table with the following data:

	Floor Requested	Task
⚙️	12	Need Materials

**Figure 10:** Employee Queue Details Page

Figure 11 shows the Admin Queue Details Page. This page is similar to the Employee Queue Details Page; however, when an admin is logged in they have the option to edit the queue.

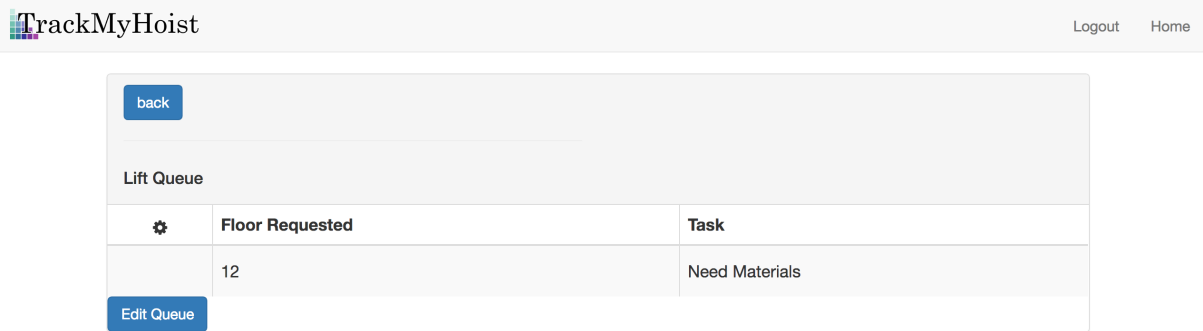
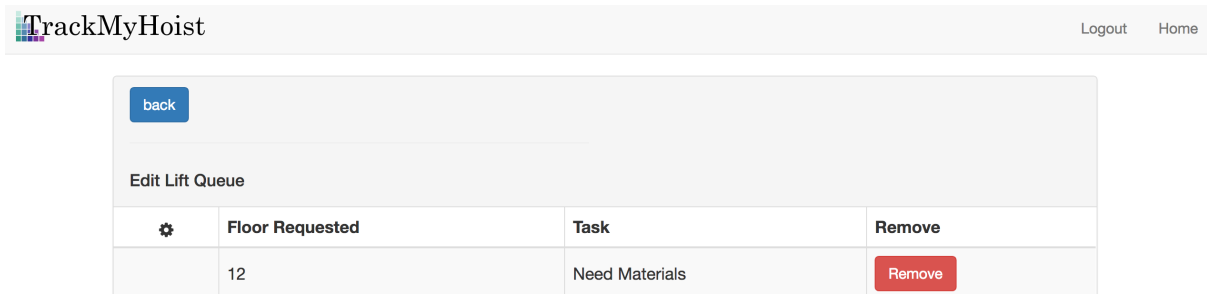


Figure 11: Admin Queue Details Page

Figure 12 shows the Admin Edit Queue Page. Admins have the option to remove tasks from the queue as they see fit.



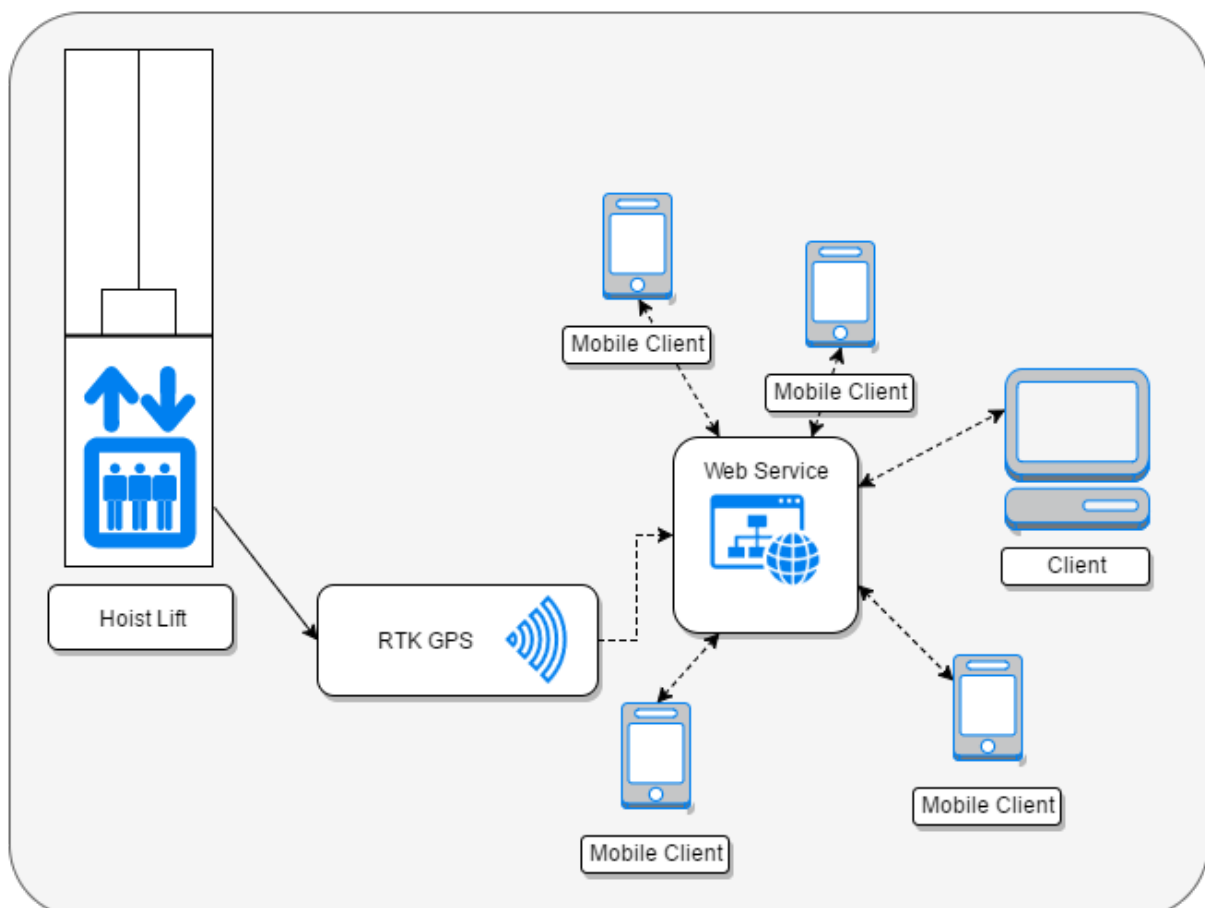
The screenshot displays the 'Admin Edit Queue Page' in the Track My Hoist application. At the top left is the 'TrackMyHoist' logo, and at the top right are 'Logout' and 'Home' links. Below the header is a 'back' button. The main content area is titled 'Edit Lift Queue' and contains a table with the following structure:

	Floor Requested	Task	Remove
	12	Need Materials	Remove

**Figure 12:** Admin Edit Queue Page

## 9 Architectural Diagram

Software projects will generally follow an architecture archetype or pre-described solution that describes the high-level functions and behaviors of the system. This section includes Figure 13, which represents the general layout and system of our design. Our solution consists of physical devices, the pair of Reach RTK GPS modules and their antennae, and software to connect multiple users to our application that monitors and displays the status of the hoist lift. From the user's perspective, the system behaves similar to a Client-Server Architecture on the top-most layer. Construction site workers will access the application as Clients and request information from the Server. Regardless of the number of users, the only interaction is between application and web service. This fits the CS description, as Clients do not communicate with each other.



**Figure 13:** Architectural Diagram

## 10 Technologies Used

This project will utilize different physical devices, in addition to a collection of computer science languages, techniques, and problem solving.

### Devices

**Reach RTK GPS by Emlid** Small yet powerful modules that work in pairs to measure and transmit accurate positional data. Each device is powered by an Intel Edison microprocessor and offers Wi-Fi connectivity to deliver data wirelessly through TCP as well as Bluetooth.

**Cloud Server** Immaterial location for storing and organizing encoder data to be read by user devices.

### Programming Languages

**HTML5** HTML5 is a Hyper Text Markup Language used for structuring and presenting content on the World Wide Web. It is the fifth and current version of the HTML standard.

**CSS** Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.

**PHP** Server communication language PHP is a server-side scripting language designed primarily for web development but is also used as a general-purpose programming language. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems and web frameworks.

### Other Technologies

**Bootstrap** Bootstrap is a front-end web framework that provides HTML- and CSS-based templates to create a responsive and professional interface.

**MySQL** MySQL is a relational database management system written in C and C++.

**Overleaf** Web-based service for collaborative editing of  $\text{\LaTeX}$  documents.

**ReachView** Browser interface for set-up and configuration of Reach RTK GPS modules.

**TCP** Transmission Control Protocol; a set of procedures for sending and receiving data over Internet Protocol networks.

**LLH** Longitude Latitude Height; a GPS output format that includes positional data for 3 dimensions.

## **11 Design Rationale**

### **11.1 Justification of User Experience**

For our solution, we implemented a hierarchical log in structure; the managers will be in charge of the main account for a given project with their own log in information, and a separate password for employees to view the project will be set up by the manager for distribution among regular employees. This structure will ensure easy access for all employees on-site while still giving control to the site manager. Additionally, the display of hoist lifts and their respective levels will reside on the home page of the application in order to reduce the steps necessary to navigate the application and simplify the user experience. The queue of requests and all information concerning a given request will be displayed on a different page that users can easily navigate to with one click; this separation of information is in order to make our webpage less cluttered and more organized.

### **11.2 Justification of Technologies Used**

For our application we are using HTML, CSS, JavaScript, PHP, and MySQL. We are using HTML as it is the standard language for web pages, as well as CSS to style our web pages. JavaScript is used to create a responsive interface and increase the ease of use of our application. We have used PHP and MySQL to save and store users' log in information and hoist lift information.

Our initial solution involved the use of multiple elements that needed to work together. The first part was a rotary encoder that needs to be attached to the shaft of the hoist lift motor. This encoder would collect the rotational data of the motor which is used to calculate the floor at which the lift is. The encoder is then connected to a Cradle Point Static GPS module. This GPS module then needs to be coupled with an Arduino based board like the Particle Board to enable cellular capabilities required to transmit the data to the servers. Complexity wise this was a lot of different pieces that had to work together for the success of the project. Also, the footprint of the GPS unit was pretty big to be installed in a construction site environment. Lastly, the cost associated with this solution was upwards of two thousand dollars per lift. Hence, it was not a feasible solution cost wise!

On further research, we were able to find a better, accurate, and cost effective solution. This solution, which is our current solution, uses the RTK GPS module - Reach RTK, developed by Emlid. It is an IoT device built with an RTK GPS module on the same chip. The Reach device collects data related to longitude, latitude, and elevation and transmits



it in real time. This implementation costs six hundred dollars for tracking the first lift and additional three hundred dollars for subsequent lifts at the same site. The footprint of this device is very small too. Hence, we decided to go ahead with this solution as it reduced the costs significantly and the concerns pertaining the installation were eliminated too.

All the data that is collected by the Reach RTK GPS module is transmitted to our servers from where it is relayed to the website providing real time updates to the construction site workers.

## 12 Test Plan

The plan outlined below illustrates the cases that we considered for testing in order to verify if the functionalities of our system are in order.

### Alpha Testing

- Log In Verification
  - Purpose: To verify if the log in aspect is functional and takes the user to the correct display
  - Assumptions: User enters the correct login information
  - Steps:
    1. Open app
    2. Enter login information and submit
    3. See page displaying correct information corresponding to login
- Real Time Testing
  - Purpose: To verify that the system updates with real-time information of what level the hoist lift is at
  - Assumptions: None
  - Steps:
    1. Perform Log In Verification
    2. Change position of lift and compare to data displayed on screen
- Request Verification
  - Purpose: To verify if requests are added to the queue
  - Assumptions: User enters correct information for request
  - Steps:
    1. Perform Log In Verification
    2. Fill out request form
    3. Submit request
    4. Verify that new request is scheduled in the queue

## **Beta Testing**

- Process Feedback
  - Purpose: To obtain feedback from potential customers
  - Assumptions: None
  - Steps:
    1. Test on hoist lift
    2. Obtain feedback from company regarding interface and functionality

## 13 Risk Analysis

The risk analysis table outlined in Figure 14 provides a summary of the various risks involved in the design and implementation of this project. The various aspects of the Risk Analysis Report are:

- Name of the Risk:
  - Suggests a short description of the risk.
- Consequences:
  - Summarizes the effects a particular risk would have on the end result of the project.
- Probability:
  - The chances of a particular risk to occur.
- Severity:
  - Lists the magnitude of seriousness of a given risk.
- Impact:
  - Product of the probability and severity of a particular risk. The table is sorted by the impact a risk would have.
- Mitigation Strategies:
  - Strategies to lessen the probability of the risk occurring or reduce the severity of the risk which would eventually reduce the impact of the risk.

Name of Risk	Consequences	Probability (0 < Probability < 1)	Severity (0 < Severity < 10)	Impact (Probability * Severity)	Two Mitigation Strategies
Misunderstanding or Changing Requirements	Leads to an incorrect implementation of the product	1	7	7	<ul style="list-style-type: none"> <li>- Have frequent meetings with client to confirm if heading on the right track</li> <li>- Make system mock ups and prototypes for the client to test</li> </ul>
Miscommunication Among Group Members	Duplication of tasks which may lead to unsuccessful completion of the project	0.5	6	3	<ul style="list-style-type: none"> <li>- Discuss every minute detail of tasks to ensure everyone is on the same page</li> <li>- Create a group text to communicate with ease</li> </ul>
Not Getting Timely Information From the Customer	Lot of assumptions leading to an end result different from what the customer asked for	0.4	6	2.4	<ul style="list-style-type: none"> <li>- Ask more questions to customer and clarify doubts</li> <li>- Show multiple mock ups to stay on track</li> </ul>
Team Member Falling Sick	More tasks for other team members to cover for the missing team member	0.3	4	1.2	<ul style="list-style-type: none"> <li>- Get good sleep</li> <li>- Visit the doctor</li> </ul>
Losing All or Parts of the Code	Rewrite all the code and may need more time for submission	0.1	9	0.9	<ul style="list-style-type: none"> <li>- Use GitHub to have timely back up of the code</li> <li>- Maintain Git Directory on individual local machines</li> </ul>
Running Out of Time	Unsuccessful completion of the project and probably a bad grade	0.05	9	0.45	<ul style="list-style-type: none"> <li>- Follow Gantt Chart and manage time</li> <li>- Start working in the tasks well in advance</li> </ul>

Figure 14: Risk Analysis Table

## 14 Development Timeline

The Development Timeline is a graphical representation of all the tasks that are needed for the completion of the project. This graphical model is known as a Gantt Chart. The Gantt Chart for our project is provided in Figure 15.

The table columns in the Gantt Chart lists the set time periods over which the project will be completed. For our project, we have divided the project across nine months. The rows of the table in the chart list the various tasks that need to be completed and the expected timeframe during which they will be worked on. The intersection of the rows and columns in the Gantt Chart will provide data concerning the team member to which a respective task will be assigned and when a particular task is due for submission or presentation. Highlighted are the deadlines for the project documents and demos.

The Gantt Chart proves to be an efficient visual method of tracking the different tasks of the project and helps staying on schedule.

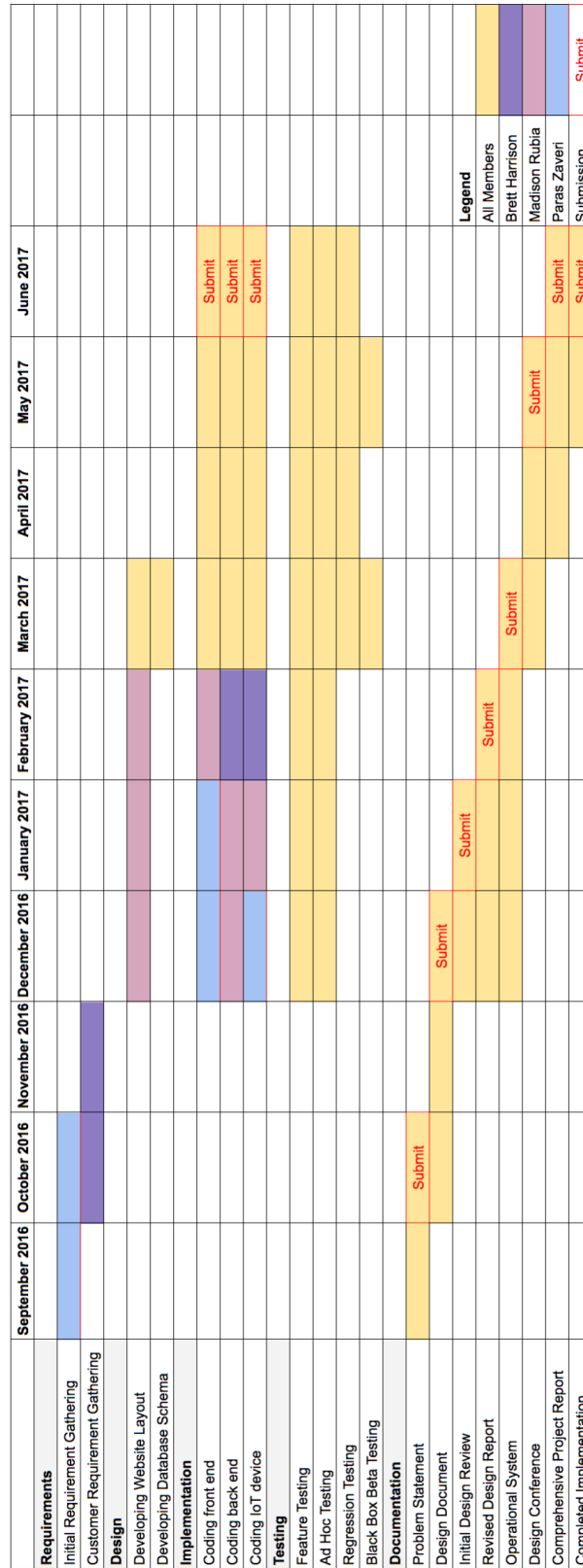


Figure 15: Gantt Chart

## **15 Societal Issues**

### **15.1 Ethical**

This project helps workers at the construction site to monitor the activity of the hoist lift and make its use efficient. This in turn helps to reduce the wait times for the people requesting these lifts, helps managers get a realistic time frame for a task to be completed when assigned to an employee, and helps build worker morale and boost employee productivity.

### **15.2 Societal**

The desired impact for this project would affect the construction site and the employees present, in addition to the companies both utilizing and renting hoist lifts. The intention is to help, adding ease and amenities to a rudimentary operation. Over time, one long-term outcome desired is a reduction in the operating costs of lifts; the effect of this outcome depends on the construction company affected.

### **15.3 Political**

We feel that our project doesn't fall in the realm of any political context and hence think that it wouldn't have any political impact towards the governed or governing.

### **15.4 Economic**

The prime goal of this project was to provide a solution that is easy to install and is not expensive. The initial solution that we had in mind consisted of a Cradle Point Static GPS unit. The cost of the device alone was around eight hundred dollars. This had to be coupled with an Arduino board like the Particle board which costs around a hundred dollars. The most expensive part in this solution was a rotary encoder which costs around a thousand dollars depending on the specifications. All in all, this solution meant an installation cost of roughly two thousand dollars per lift at the site.

By conducting some more research, we came across a better and accurate solution for our project. Usually better and accurate both translate to more money, however in our case, we were fortunate enough to find a device that was almost a quarter the cost of our initial solution and much more accurate, with a much smaller foot print. The RTK GPS



solution that we provide would cost the construction site six hundred dollars for the first lift and three hundred dollars for any subsequent lifts added to the site.

As one can notice, the cost savings is tremendous and that would help construction site save money.

### **15.5 Health and Safety**

Our project involves a small circuit board coupled with an antenna that would sit on top of a lift car. Apart from the usual Health and Safety regulations governing electronic devices, our project doesn't have any safety concerns associated with the operation of the device.

### **15.5 Manufacturability**

The solution that we have designed relies on the RTK GPS. Currently this device is manufactured by a company called Emlid. As technology progresses, there is a bright possibility of the cost of these devices coming down. Right from its inception to now, there has not been a shortage of availability of these devices. Hence, we are pretty optimistic that one would not run into issues concerning manufacturability of this device.

### **15.6 Sustainability**

Our solution is based on a device that runs on electricity, and has a very small physical footprint. Also, it is a device manufactured on a chip, so as the technology advances, the chances of the device having an even smaller footprint is high. We feel our solution is sustainable in the narrow and the broad sense as technology advancements would help bring the costs down and is a long term plug and play solution which can be installed and removed and reinstalled easily.

### **15.7 Environmental Impact**

As mentioned above, our solution requires power to operate. The power source can have a varying degree of impact on the environment. Power at construction sites is commonly produced on-site with generators that run on fossil fuels and emit greenhouse gases. Ideally, our solution would be powered by small batteries; therefore the environmental impact of our solution should include the environmental issues of using disposable or rechargeable lithium-ion batteries.

## **15.8 Usability**

Usability/intuitive was our primary goal when we thought of designing a solution addressing the problem at the construction site. The RTK GPS device is very easy to set up at the construction site. Once done, its is pretty straightforward to access the hoist lift data as far as one has the credentials to the web-application. The difficulty level of using the application as same as any social media or e-commerce website.

## **15.9 Lifelong learning**

This project gave us a chance to understand a part of the construction industry and look at it more closely. This involved researching and studying how hoists lift work, what kind of tasks depend on it, how personnel at the site monitor it, how the costs involved go up or down based on the lifts operation, etc. Even deciding on the device that would be most suitable in terms of cost and convenience we had to explore unknown territory. Overall, as a team, we feel that this project enabled us to learn new things as technology advances, we will be investing more time to keep learning on how we can have a better implementation of our solution.

## **15.10 Compassion**

The focus of this project was not to relieve suffering, because the problem chosen was not a cause of suffering. Looking at construction sites, workers do not suffer from the operation of hoist lifts. However, this does not mean lifts are a source of frustration or inefficiencies, and our solution can relieve these issues.

## **16 Conclusion**

Our solution provides construction companies with an easy-to-use application that is available on both desktop and mobile devices. This application allows employees to more efficiently communicate their task schedules with their manager, and allows the manager to more efficiently oversee activity on-site, thus aiding the company in cutting financial loss due to wasted time.

## Appendix

### A. Project Files

**Listing 1:** Lift Display Page

```

<!-- check if user is logged in to access page -->
<?php
    if(!session_id())
        session_start();
    if(isset($_SESSION['logon'])) {
        if($_SESSION['logon'] == 2) {
            header("location:home.php");
            die();
        }
        else if($_SESSION['logon'] == 0) {
            header("location:index.php");
            die();
        }
    }
    else {
        header("location:index.php");
    }
?>

<?php include "header.php"; ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN//">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Admin Page</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="styles.css">
    <script src="https://code.jquery.com/jquery-3.2.1.min.js"
        integrity="sha256-hwg4gsxgFZhOsEEamdOYGBf13FyQuiTwlAQgxVSNgt4="

```

```
        crossorigin="anonymous"></script>
</head>

<script>
$(document).ready(function () {
    $('[data-toggle=offcanvas]').click(
        function () {
            $('.row-offcanvas').toggleClass('active');
        }
    )
});
</script>

<body>
<div class="container-fluid">
    <div class="row row-offcanvas row-offcanvas-left">
        <div class="col-xs-6 col-sm-3 sidebar-offcanvas" id="sidebar"
            role="navigation">
            <div class="sidebar-nav">
                <ul class="nav">
                    <li><a href="register_lift.php">Register Lift
                    </a></li>
                    <li class="nav-divider"></li>
                    <li class="active"><a href="request.php">
                    Request Lift </a>
                    </li>
                    <li class="nav-divider"></li>
                </ul>
            </div>
        </div>
        <div class="col-xs-12 col-sm-9">
            <br>
            <div class="jumbotron" id="info_field">
                <a href="#" class="visible-xs" data-toggle="offcanvas"><i
```

```

class="fa fa-lg fa-reorder"></i></a>

<?php
    include "get_num_lifts.php";
    if($num_lifts > 1)
        echo "<p id='lift_total '>".$num_lifts." Lifts </p>";
    else
        echo "<p id='lift_total '>".$num_lifts." Lift </p>";
    //dynamically generate sections for each lift in table
    for($i = 1; $i <= $num_lifts; $i++) {
        echo "<p>Lift ". $i."
            <p>Current Floor: <span class="
            'floor_monitor'.">--
            </span></p>
            <p>Destination: <span class="
            'destination_monitor'
            .">--</span></p>
            <p>Current Status: <span class="
            'status_monitor'.">--
            </span></p>
            <p>Current Task: <span class="
            'task_monitor'.">--
            </span></p>";
    }
?>
</div>
<div class="row">
    <div class="col-6 col-sm-6 col-lg-4">

        <?php
            //goal: generate info for lifts

            include_once "db_config.php";

            include "get_num_lifts.php";

```

```

        //dynamically generate sections for each lift
        //in table
        for($i = 1; $i <= $num_lifts; $i++) {
            echo "
            <h2>Lift ".$i."</h2>
            <p>Lift ".$i." Real Time Details</p>
            <p><a class='btn btn-default '
            href='view_queue.php?num="
            .$i.'">View Queue </a></p>";
        }
    ?>
</div>
</div>
<!--/row-->
</div>
<!--/span-->

</div>
</body>
<?php
    /* Add JavaScript code to the webpage for each individual lift.
    * Each block creates an XMLHttpRequest against the lift table
    * in the database to fetch and update the display. */
    include "get_num_lifts.php";
    for($i = 1; $i <= $num_lifts; $i++) {
        $j = $i - 1;
        echo "<script>
            var i = $i;
            function liftUpdate$i(){
                if (window.XMLHttpRequest) {
                    floor_request = new XMLHttpRequest();
                    destination_request = new XMLHttpRequest();
                    status_request = new XMLHttpRequest();

```

```
task_request = new XMLHttpRequest();
} else {
    floor_request = new ActiveXObject(".
'Microsoft.XMLHTTP'");
    destination_request = new ActiveXObject(".
'Microsoft.XMLHTTP'
.");
    status_request = new ActiveXObject(".
'Microsoft.XMLHTTP'");
    task_request = new ActiveXObject(".
'Microsoft.XMLHTTP'");
}
floor_request.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        document.getElementsByClassName(".
'floor_monitor'").
        [$j].innerHTML = this.responseText;
    }
};
var data = ". 'field=floor&lift_num="'. "+i".";
floor_request.open(". 'POST' ". , ".
'ajax_fetch.php' ". , true);
floor_request.setRequestHeader(".
'Content-type' ". , ".
'application/x-www-form-urlencoded' ".);
floor_request.send(data);

destination_request.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
        document.getElementsByClassName(".
'destination_monitor' ".
        ")[$j].innerHTML = this.responseText;
    }
};
var info = ". 'field=destination&lift_num="'. "+i".";
```



```
destination_request.open( ". 'POST' ". , ".
'ajax_fetch.php' '.
", true );
destination_request.setRequestHeader( ".
'Content-type' ". , ".
'application/x-www-form-urlencoded' ". );
destination_request.send( info );

status_request.onreadystatechange = function () {
    if ( this.readyState == 4 && this.status == 200 ) {
        document.getElementsByClassName( ".
        'status_monitor' '. )
        [ $j ].innerHTML = this.responseText;
    }
};
var quote = ". 'field=status&lift_num=' ". + i ". ";
status_request.open( ". 'POST' ". , ".
'ajax_fetch.php' ". , true );
status_request.setRequestHeader( ". 'Content-type' ".
", ".
'application/x-www-form-urlencoded' ". );
status_request.send( quote );

task_request.onreadystatechange = function () {
    if ( this.readyState == 4 && this.status == 200 ) {
        document.getElementsByClassName( ".
        'task_monitor' '. )
        [ $j ].innerHTML = this.responseText;
    }
};
var update = ". 'field=task&lift_num=' ". + i ". ";
task_request.open( ". 'POST' ". , ". 'ajax_fetch.php' ".
", true );
task_request.setRequestHeader( ". 'Content-type' ".
", ".
```

```
        'application/x-www-form-urlencoded' .");
        task_request.send(update);
    }
    setInterval(liftUpdate$i, 3000);
</script>";
}
?>

</html>
```

**Listing 2:** Fetch Request

```
<?php

session_start();

include_once "db_config.php";

//connect to mysql server
$conn = new mysqli($db_host, $username, $password, $dbname)
    or die("Error" . mysqli_error($conn));

$field = $_POST["field"];
$lift_num = $_POST["lift_num"];
$email = $_SESSION['email'];
$table_query = "SELECT company FROM $login_info WHERE
email='$email'";
$result = mysqli_query($conn, $table_query);
$company_name = mysqli_fetch_row($result);
$company_name = mb_strtolower($company_name[0]);

if($field == "init"){
    $query = "SELECT * FROM $company_name ORDER BY lift_num ASC";
    $result = mysqli_query($conn, $query) or
    die(mysqli_error($conn));
    $rows = mysqli_num_rows($result);
```

```
        echo $rows;
    }
    elseif($field == "floor"){

        $query = "SELECT cur_floor FROM $company_name WHERE
lift_num=$lift_num ";
        $result = mysqli_query($conn, $query) or
die(mysqli_error($conn));

        $row = mysqli_fetch_row($result);

        echo $row[0];
    }
    elseif($field == "status"){

        $query = "SELECT cur_task FROM $company_name
WHERE lift_num=$lift_num ";
        $result = mysqli_query($conn, $query) or
die(mysqli_error($conn));

        $row = mysqli_fetch_row($result);

        echo $row[0];
    }
    elseif($field == "destination"){

        $query = "SELECT qfn1 FROM $company_name WHERE
lift_num=$lift_num ";
        $result = mysqli_query($conn, $query) or
die(mysqli_error($conn));

        $row = mysqli_fetch_row($result);

        echo $row[0];
    }
}
```

```
elseif($field == "task"){

    $query = "SELECT qtask1 FROM $company_name
    WHERE lift_num=$lift_num ";
    $result = mysqli_query($conn, $query) or
    die(mysqli_error($conn));

    $row = mysqli_fetch_row($result);

    echo $row[0];
}
else
    echo "Error";

?>
```

**Listing 3:** TCP Client

```
<?php
/* This program connects to the Emlid Reach module currently
outputting positional data as a TCP Server. The data is received
as an LLH segment and is parsed to select the relevant data.
Before this data is sent to the lift table on the data base,
the current floor, speed, and direction are computed. */

include "sock_config.php";
//Create connection
$socket = socket_create(AF_INET, SOCK_STREAM, 0) or
die("Socket creation failure\n");

$result = socket_connect($socket, $host, $port) or
die("Server connection fail\n");

socket_write($socket, $message, strlen($message)) or
die("Message failure\n");
```

```
//Base initialization of variable that stores current time
$time1 = 0;

//Read LLH from server
$data = socket_read($socket, 1024) or die("No server response\n");
//Split the data
$list = explode(" ", $data);
//Split the timestamp
list($year, $month, $day) = explode("/", $list[0]);
//Save the current timestamp
$time2 = $time1;
//Get a new timestamp
$time1 = explode(":", $list[1]);
//Set data to be sent
$lat = $list[4];
$long = $list[5];
//Save the current height
$height1 = 0;
$height2 = $height1;
//Save the next height
//The position of the height value in $list can vary following the
//explode() call
if($list[8] != '')
    $height1 = $list[8];
elseif($list[9] != '')
    $height1 = $list[9];
elseif($list[10] != '')
    $height1 = $list[10];
else
    $height1 = 0;
//Hard-coded database and table information
$company_name = "scu";
$lift = 1;
$floor_height = 3; //default height in meters
```

```
//Connect to database
include "db_config.php";
$conn = mysqli_connect($db_host, $username, $password, $dbname)
        or die("Error" . mysqli_error($conn));

//Get the actual floor height
$query = "SELECT floor_height FROM $company_name WHERE
lift_num=$lift ";

$result = mysqli_query($conn, $query);

if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        $floor_height = $row["floor_height"];
    }
} else {
    echo "0 results";
}

//Calculate the current floor of the lift
if($floor_height > 0)
    $floor = ($height1/$floor_height);

include "gps_utility.php";
//Calculate speed
$dist = $height1 - $height2;
$split = subTime($time2[0], $time2[1], $time2[2], $time1[0],
    $time1[1], $time1[2]);
$speed = $dist/$split;
//Determine direction of motion
if($speed > 5) //Unit is m/s
    $state = "up";
elseif($speed < 5 && $speed > -5)
    $state = "stop";
else
```

```
        $state = "down";
//Update entries in lift table
$query = "UPDATE $company_name SET lift_height=$height1,
cur_floor=$floor, speed=$speed WHERE lift_num=$lift ";
echo $query;
mysqli_query($conn, $query) or die(mysqli_error($conn));

mysqli_close($conn);
//Reload and get new data
$url1=$_SERVER['REQUEST_URI'];
header("Refresh: 1; URL=$url1 ");
socket_close($socket);
?>
```

**Listing 4:** Database Login Information

```
<?php

$db_host = "localhost";
$username = "root";
$password = "";
$dbname = "sdb_mrubia";
$login_info = "login_info";

?>
```

**Listing 5:** Edit Lift Details

```
<?php

session_start();

include_once "db_config.php";

//connect to mysql server
$conn = new mysqli($db_host, $username, $password, $dbname)
        or die("Error" . mysqli_error($conn));
```

```
?>
```

**Listing 6:** Edit Lift Queue

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN/"
<html xmlns="http://www.w3.org/1999/xhtml">

<!-- check if user is logged in to access page -->
<?php
    if(!session_id()) session_start();
    if(isset($_SESSION['logon'])) {
        if($_SESSION['logon'] == 2) {
            header("location:home.php");
            die();
        }
        else if($_SESSION['logon'] == 0) {
            header("location:index.php");
            die();
        }
    }
    else {
        header("location:index.php");
    }
?>

<head>
    <title>Home Page For Project Site [Project Name]</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/3.3.7/css/bootstrap.min.css">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/
4.4.0/css/font-awesome.min.css" rel='stylesheet' type='text/css'>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>

<script>
$(document).ready(function() {
```



```
    $ ('[ data-toggle=offcanvas ] '). click (function () {
        $ ('.row-offcanvas '). toggleClass (' active ');
    });
</script>

<?php include "header.php"; ?>

<body>

<div class="col-md-10 col-md-offset-1">
    <div class="panel panel-default panel-table">
        <div class="panel-heading">
            <div class="row">
                <div class="col col-xs-6">

<?php
    if (isset($_SESSION['logon']) && isset(
        $_SESSION['num'])) {
        if ($_SESSION['logon'] == 1) {
            echo "
                <a class='btn btn-primary'
                href='/~mrubia/TrackMyHoist/
                view_queue.php?num=".
                $_SESSION['num'] .
                "'>back</a>
            ";
        }
        if ($_SESSION['logon'] == 2) {
            echo "
                <a class='btn btn-primary'
                href='home.php'>back</a>
            ";
        }
    } ?>

<hr>
<h3 class='panel-title '>Edit Lift Queue</h3>
```

```
</div></div></div>
```

```
<?php
```

```
    include "db_config.php";

    //connect to mysql server
    $conn = new mysqli($db_host, $username, $password, $dbname)
        or die("Error" . mysqli_error($conn));

    //get company name
    if (isset($_SESSION['email'])) {
        $query = "SELECT company FROM $login_info WHERE email = '".
            $_SESSION['email']."'";
        $result = mysqli_query($conn, $query);
    }

    while($row = mysqli_fetch_array($result)) {
        $company = $row['company'];
    }

    if (isset($_SESSION['num'])) {

        $query = "SELECT * FROM $company WHERE lift_num = '".
            $_SESSION['num']."'";
        $result = mysqli_query($conn, $query);

        while($row = mysqli_fetch_array($result)) {
            $qfn1 = $row['qfn1'];
            $qtask1 = $row['qtask1'];
            $qfn2 = $row['qfn2'];
            $qtask2 = $row['qtask2'];
            $qfn3 = $row['qfn3'];
            $qtask3 = $row['qtask3'];
            $qfn4 = $row['qfn4'];
```

```

    $qtask4 = $row[ 'qtask4 '];
    $qfn5 = $row[ 'qfn5 '];
    $qtask5 = $row[ 'qtask5 '];
    $qfn6 = $row[ 'qfn6 '];
    $qtask6 = $row[ 'qtask6 '];
    $qfn7 = $row[ 'qfn7 '];
    $qtask7 = $row[ 'qtask7 '];
    $qfn8 = $row[ 'qfn8 '];
    $qtask8 = $row[ 'qtask8 '];
}
echo "
<div class='panel-body'>
  <table class='table table-striped
table-bordered table-list '>
  <thead>
  <tr>
    <th><em class='fa fa-cog'></em></th>
    <th>Floor Requested</th>
    <th>Task</th>
    <th>Remove</th>
  </tr>
</thead>
<tbody>
  <tr>
";

for($i = 1; $i < 9; $i++) {
  if("${qfn" . $i} == NULL) {
    continue;
  }
  echo"
    <td>
      <td>".${"qfn" . $i}."</td>
      <td>".${"qtask" . $i}."</td>
      <td><a class='btn btn-danger'

```

```

        href='edit_queue.php?order="'.
        $i.'">Remove</a></td>
            </td>
                ";
    }
    echo "</table>";
}

if(isset($_GET['order'])) {
    //remove task from queue
    $query = "UPDATE $company SET qfn".$_GET['order']."' = NULL, qtask".$_GET['order']."' = NULL WHERE lift_num = '".$_SESSION['num']."' ";
    $result = mysqli_query($conn,$query);

    if(isset($_SESSION['num'])) {
        header("location:~/mrubia/TrackMyHoist/view_queue.php?num=".$_SESSION['num']);
    }
}

?>

</body>
</html>

```

**Listing 7:** Returns number of current lifts

```

<?php
include_once "db_config.php";

//connect to mysql server
$conn = new mysqli($db_host, $username, $password, $dbname)
    or die("Error" . mysqli_error($conn));

//get company name
if(isset($_SESSION['email'])) {

```

```

$query = "SELECT company FROM $login_info WHERE email = '".
$_SESSION['email']."'";
$result = mysqli_query($conn,$query);
}
while($row = mysqli_fetch_array($result)) {
    $company = $row['company'];
}

//get number of rows in table i.e. how many lifts are registered
$query = "SELECT lift_num from $company";
$result = mysqli_query($conn,$query);

$num_lifts = mysqli_num_rows($result);
?>

```

**Listing 8:** Utility Calculation Functions

```
<?php
```

```

function roverDistance($lat1 , $lon1 , $lat2 , $lon2){
    $lat1 = $lat1 * M_PI / 180.0;
    $lon1 = $lon1 * M_PI / 180.0;

    $lat2 = $lat2 * M_PI / 180.0;
    $lon2 = $lon2 * M_PI / 180.0;

    // radius of earth in meters
    $r = 6378100;

    // P
    $rho1 = $r * cos($lat1);
    $z1 = $r * sin($lat1);
    $x1 = $rho1 * cos($lon1);
    $y1 = $rho1 * sin($lon1);

    // Q

```

```

        $rho2 = $r * cos($lat2);
        $z2 = $r * sin($lat2);
        $x2 = $rho2 * cos($lon2);
        $y2 = $rho2 * sin($lon2);

        // Dot product
        $dot = ($x1 * $x2 + $y1 * $y2 + $z1 * $z2);
        $cos_theta = $dot / ($r * $r);

        $theta = acos($cos_theta);

        // Distance in Meters
        return $r * $theta;
    }

function subTime($h1, $m1, $s1, $h2, $m2, $s2){
    return ($h2-$h1)*3600 + ($m2-$m1)*60 + ($s2-$s1);
}
roverDistance(121, 37, 120, 36);
subTime(1, 23, 10, 1, 56, 45);

?>

```

**Listing 9:** Website Banner

```

<?php //session_start(); ?>
<nav class="navbar navbar-default">
    <div class="container-fluid">
        
        <ul class="nav navbar-nav navbar-right">
            <?php if(isset($_SESSION['logon'])) {
                //if logged in, can log out
                if($_SESSION['logon'] == 1 ||
                    $_SESSION['logon'] == 2)

```

```
        {
            echo "
<li><a id='logout' class='active' href='/~mr
TrackMyHoist/logout.php'>Logout</a></li>
"; }

        //admin logged in
        if($_SESSION['logon'] == 1) {
            echo "
<li><a id='home' class='active' href='/~mr
TrackMyHoist/admin_home.php'>Home</a></li> "
            //user logged in
            else if($_SESSION['logon'] == 2)
                echo "
<li><a id='home' class='active' href='/~mr
TrackMyHoist/home.php'>Home</a></li> "; }
            else {
                echo "
<li><a id='home' class='active' href='/~mr
TrackMyHoist/index.php'>Home</a></li>
<li><a id='login' class='active' href='/~mr
TrackMyHoist/login.php'>Login</a></li> ";
                }
        }
    }

    else {
        //user logged out, no home page
        echo "

<li><a id='home' class='active' href='index
Home</a></li>
<li><a id='login' class='active' href='logi
Login</a></li> ";
        }
    }
?>
</ul>
</div>
```

```
</nav>
```

**Listing 10:** Home Page

```
<!-- check if user is logged in to access page -->
<?php
    if (!session_id()) session_start();
    if (isset($_SESSION['logon'])) {
        if ($_SESSION['logon'] == 1) {
            header("location:admin_home.php");
            die();
        }
        else if ($_SESSION['logon'] == 0) {
            header("location:index.php");
            die();
        }
    }
    else {
        header("location:index.php");
    }
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN/"
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <title>Home Page For Project Site [Project Name]</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>

<script>
//$(document).ready(function() {
//    $('[data-toggle=offcanvas]').click(function() {
//        $('.row-offcanvas').toggleClass('active');

```



```
//   });
</script>

<?php include "header.php"; ?>

<body>
<div class="container-fluid">
  <div class="row row-offcanvas row-offcanvas-left">
    <div class="col-xs-6 col-sm-3 sidebar-offcanvas" id="sidebar"
    role="navigation">
      <div class="sidebar-nav">
        <ul class="nav">

          <li class="active"><a href="request.php">
            Request Lift </a>
          </li>

        </ul>
      </div>
      <!--/.well -->
    </div>
    <!--/span-->

    <div class="col-xs-12 col-sm-9">
      <br>
      <div class="jumbotron">
        <a href="#" class="visible-xs" data-toggle="offcanvas"><i
        class="fa fa-lg fa-reorder"></i></a>

        <p>Info About Lift Displays</p>
      </div>
      <div class="row">
        <div class="col-6 col-sm-6 col-lg-4">
          <?php
            include_once "db_config.php";
```

```

//connect to mysql server
$conn = new mysqli($db_host, $username, $password,
    $dbname)
    or die("Error" . mysqli_error($conn));

//get company name
if (isset($_SESSION['email'])) {
    $query = "SELECT company FROM $login_info WHERE
    email = '".$_SESSION['email']."'";
    $result = mysqli_query($conn,$query);
}
while($row = mysqli_fetch_array($result)) {
    $company = $row['company'];
}

//get number of rows in table i.e. how many lifts
//are registered
$query = "SELECT lift_num from $company";
$result = mysqli_query($conn,$query);

$num_lifts = mysqli_num_rows($result);

//dynamically generate sections for each lift
for($i = 1; $i <= $num_lifts; $i++) {
    echo "
    <h2>Lift ".$i."</h2>
    <p>Lift ".$i." Real Time Details</p>
    <p><a class='btn btn-default '
    href='view_queue.php?num=".$i."'>
    View Queue Âž</a></p>";
}
?>
</div>
</div>
<!--/row-->

```

```
</div>
<!--/span-->
```

```
</div>
</div>
```

```
</body>
</html>
```

**Listing 11: Index**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN/"
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
  />
  <link rel="stylesheet" href="http://www.w3schools.com/lib/w3.css">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/3.3.7/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="styles.css">

</head>

<?php include "header.php"; ?>

<body>

<!--
  <div id="login_section" >
    <p><a class="active" href="register.php">REGISTER PROJECT</a></p>
    <p><a class="active" href="login.php">LOGIN</a></p>
  </div>
-->
```

```

<div class="container">
  <div class="row">
    <div class="social-box">
      <div class="row mg-btm">
        <div class="col-md-12">
          <a href="register.php" style="width:100px; margin-bottom: 10px;
margin-top:10em;"
          class="btn btn-success center-block btn-block">
            Register</a>
        </div>
      </div>
    </div>
    <div class="row">
      <div class="col-md-12">
        <a href="login.php" style="width:100px"
          class="btn btn-primary center-block btn-block">Login</a>
      </div>
    </div>
  </div>
</div>
</body>
</html>

```

**Listing 12:** Login Page

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN/"
<html xmlns="http://www.w3.org/1999/xhtml">

<?php

    //start session
    //session_start();

    ob_start();

    $emailErr = $pswErr = "";

```

```
if($_SERVER["REQUEST_METHOD"] == "POST")
{
    include "db_config.php";

    //connect to mysql server
    $conn = new mysqli($db_host, $username, $password, $dbname)
        or die("Error" . mysqli_error($conn));

    $email = $_POST['email'];
    $psw = $_POST['password'];

    //protect from mysql injection
    $email = stripslashes($email);
    $psw = stripslashes($psw);
    $email = mysqli_real_escape_string($conn, $email);
    $psw = mysqli_real_escape_string($conn, $psw);

    //session var to check if user is logged in
    //0 is not logged in, 1 is admin user, and 2 is regular
    //employee
    $_SESSION['logon'] = 0;

    //for errors
    $action = array();
    $action['res'] = NULL;

    ////////////////////////////////////////
    //EMPTY FIELDS
    if(empty($email)) {
        $action['res'] = 'error';
        $emailErr = "Please Enter Email";
    }
    if(empty($psw)) {
        $action['res'] = 'error';
```

```
        $pswErr = "Please Enter Password";
    }
    ///////////////////////////////////////////////////////////////////

//if no errors
if($action['res'] != 'error') {
    //check database
    $query = "SELECT * FROM $login_info WHERE email = '$email'
AND password = '$psw'";
    $result = mysqli_query($conn,$query);

    // login success
    if(mysqli_num_rows($result) == 1)
    {
        $_SESSION['email'] = $email;
        $row = mysqli_fetch_array($result);
        //if admin, redirect to admin home page
        //register global var admin
        //session_register("admin");
        //header("location:admin_home.php");
        $_SESSION['logon'] = 1;
        header("location:admin_home.php");
        //else redirect to employee home page
    }
    else
    {
        // if employee is logging in
        $query = "SELECT * FROM $login_info WHERE
email = '$email' AND sub_password = '$psw'";
        $result = mysqli_query($conn,$query);

        // login success
        if(mysqli_num_rows($result) == 1)
        {
            $_SESSION['email'] = $email;
```

```
                $_SESSION['logon'] = 2;
                header("location:home.php");
            }
        }
    }
?>

<head>
    <title >Login</title >
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>

<?php include 'validate_login.php'; ?>

<?php include "header.php"; ?>

<body>
<div class="container">
    <div class="row">
        <div class="col-md-6 col-md-offset-3">
            <div class="panel panel-login">
                <div class="panel-heading">
                    <div class="row">
                        <div class="col-xs-6">
                            <a class='btn btn-primary' href='index.php'>back</a>
                            <br>
                            <hr style="width:37em">
                            <a href="login.php" class="active" id="login-form-link">
                                Login</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
<hr style="width:37em">
</div>
<div class="panel-body">
  <div class="row">
    <div class="col-lg-12">
      <form id="login-form" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="post"
role="form" style="display: block;">
        <div class="form-group">
          <span class="error"><?php echo $emailErr;?></span>
          <input type="text" name="email" id="email" tabindex="1"
            class="form-control" placeholder="Email" value="">
        </div>
        <div class="form-group">
          <span class="error"><?php echo $pswErr;?></span>
          <input type="password" name="password" id="password"
            tabindex="2" class="form-control" placeholder="Password">
        </div>
        <div class="form-group">
          <div class="row">
            <div class="col-sm-6 col-sm-offset-3">
              <input type="submit" name="submit" id="login-submit"
                tabindex="4" class="form-control btn btn-primary"
                value="Log In">
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>
</div>
</div>
```



```
</body>
</html>
```

**Listing 13:** Logout Page

```
<?php

    //start session
    session_start();
    if (isset($_SESSION['logon'])) {
        session_unset($_SESSION['logon']);
        session_unset($_SESSION['num']);
        session_unset($_SESSION['company']);
        $_SESSION['logon'] = 0;
    }
    session_destroy();
    header("location: index.php");

?>
```

**Listing 14:** Registration Page

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN/"
<html xmlns="http://www.w3.org/1999/xhtml">

<?php

    session_start();

    ob_start();

    $projectErr= $companyErr = $nameErr = $emailErr = $pswErr =
    $subpswErr = "";

    if ($_SERVER["REQUEST_METHOD"] == "POST")
    {
        include "db_config.php";
```

```
//connect to mysql server
$conn = new mysqli($db_host, $username, $password, $dbname)
        or die(" Error" . mysqli_error($conn));

////////////////////////////////////
//collect info from form
$project = $_POST['project'];
$company = $_POST['company'];
$name = $_POST['name'];
$email = $_POST['email'];
$psw = $_POST['password'];
$subpsw = $_POST['sub_password'];
$admin = 1;

$project = stripslashes($project);
$company = stripslashes($company);
$name = stripslashes($name);
$email = stripslashes($email);
$psw = stripslashes($psw);
$subpsw = stripslashes($subpsw);
$project = mysqli_real_escape_string($conn, $project);
$company = mysqli_real_escape_string($conn, $company);
$name = mysqli_real_escape_string($conn, $name);
$email = mysqli_real_escape_string($conn, $email);
$psw = mysqli_real_escape_string($conn, $psw);
$subpsw = mysqli_real_escape_string($conn, $subpsw);

////////////////////////////////////

//for errors
$action = array();
$action['res'] = NULL;

//INCORRECT FIELDS
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
```

```
        $action['res'] = 'error';
        $emailErr = "Please Enter Valid Email";
    }
    //psw must be 8 characters or longer
    if(strlen($psw) < 8) {
        $action['res'] = 'error';
        $pswErr = "Password Must Be At Least 8 Characters";
    }

    $query = "SELECT * FROM $login_info WHERE email = '$email'";
    $result = mysqli_query($conn,$query);

    //email already exists
    if(mysqli_num_rows($result) == 1) {
        $action['res'] = 'error';
        $emailErr = "Email Already Registered";
    }

    //////////////////////////////////////
    //EMPTY FIELDS
    if(empty($project)) {
        $action['res'] = 'error';
        $projectErr = "Please Enter Project Name";
    }
    if(empty($company)) {
        $action['res'] = 'error';
        $companyErr = "Please Enter Company Name";
    }
    if(empty($name)) {
        $action['res'] = 'error';
        $nameErr = "Please Enter Company Name";
    }
    if(empty($email)) {
        $action['res'] = 'error';
```

```
        $emailErr = "Please Enter Email";
    }
    if(empty($psw)) {
        $action['res'] = 'error';
        $pswErr = "Please Enter Password";
    }
    if(empty($subpsw)) {
        $action['res'] = 'error';
        $subpswErr = "Please Enter Employee Password";
    }

    ///////////////////////////////////////////////////////////////////

// if no errors
if($action['res'] != 'error') {
    //insert login info into database
    $query = "INSERT INTO $login_info (project , company,
name, email, password, admin, sub_password) VALUES
('$project' , '$company' , '$name' , '$email' , '$psw' ,
'$admin' , '$subpsw')";
    $result = mysqli_query($conn,$query);

    //create lift list table for company
    $query2 = "CREATE TABLE ".$company." ( lift_num INT(2),
cur_floor INT(3), cur_task VARCHAR(50), qfn1 INT(3),
qtask1 VARCHAR(50),
qfn2 INT(3), qtask2 VARCHAR(50),
qfn3 INT(3), qtask3 VARCHAR(50),
qfn4 INT(3), qtask4 VARCHAR(50),
qfn5 INT(3), qtask5 VARCHAR(50),
qfn6 INT(3), qtask6 VARCHAR(50),
qfn7 INT(3), qtask7 VARCHAR(50),
qfn8 INT(3), qtask8 VARCHAR(50)
)";
    $result2 = mysqli_query($conn,$query2);
```

```
                header("location: successful_register.php");
                session_destroy();
            }
        }
        ob_end_flush();
    ?>

<head>
    <title >Register Company</title >
    <meta http-equiv="Content-Type" content="text/html;
    charset=utf-8" />
    <link rel="stylesheet"
    href="http://www.w3schools.com/lib/w3.css">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
    bootstrap/3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="styles.css">

</head>

<?php include "header.php"; ?>

<body>

<div class="container">
    <div class="row">
        <div class="col-md-6 col-md-offset-3">
            <div class="panel panel-login">
                <div class="panel-heading">
                    <div class="row">
                        <div class="col-xs-6">
                            <a class='btn btn-success' href='index.php'>back</a>
                            <br>
                            <hr style="width:37em">
                            <br>

```

```
</div>
</div>
<div class="row">
  <div class="col-xs-6">
    <a href="register.php" class="active" id="login-form-link">
      Register Project </a>
    </div>
  </div>
  <hr style="width:37em">
</div>
<div class="panel-body">
  <div class="row">
    <div class="col-lg-12">
      <form id="login-form" action="<?php
echo htmlspecialchars($_SERVER["PHP_SELF"]);?>" method="post"
role="form" style="display: block;">
        <div class="form-group">
          <span class="error"><?php echo $projectErr;?></span>
          <input type="text" name="project" id="project" tabindex="1"
            class="form-control" placeholder="Project Name" value="">
        </div>
        <div class="form-group">
          <span class="error"><?php echo $companyErr;?></span>
          <input type="text" name="company" id="company" tabindex="2"
            class="form-control" placeholder="Company">
        </div>
        <div class="form-group">
          <span class="error"><?php echo $nameErr;?></span>
          <input type="text" name="name" id="name" tabindex="2"
            class="form-control" placeholder="Name">
        </div>
        <div class="form-group">
          <span class="error"><?php echo $nameErr;?></span>
          <input type="text" name="email" id="email" tabindex="2"
            class="form-control" placeholder="Email">
        </div>
      </div>
    </div>
  </div>
</div>
```

```
</div>
<div class="form-group">
  <span class="error"><?php echo $pswErr;?></span>
  <input type="text" name="password" id="password" tabindex="2"
    class="form-control" placeholder="Admin Password*">
</div>
<div class="form-group">
  <span class="error"><?php echo $subpswErr;?></span>
  <input type="text" name="sub_password" id="sub_password"
    tabindex="2" class="form-control"
    placeholder="Employee Password**">
</div>
<div class="form-group">
  <div class="row">
    <div class="col-lg-12">
      <span style="color:gray; font-size: .8em;">
        <p>*Admin Password is the main password for admins to
          manage the project</p>
        </p>**Employee Password is the general password for all
          regular employees to login</p>
      </span>
    </div>
  </div>
</div>
<div class="form-group">
  <div class="row">
    <div class="col-sm-6 col-sm-offset-3">
      <input type="submit" name="submit" id="login-submit"
        tabindex="4" class="form-control btn btn-success"
        value="Register">
    </div>
  </div>
</div>
</form>
</div>
```

```

    </div>
  </div>
</div>
</div>
</div>
</div>
</body>
</html>

```

**Listing 15: Add Lift Page**

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN/"
<html xmlns="http://www.w3.org/1999/xhtml">

<?php
  if(!session_id()) session_start();
  if(isset($_SESSION['logon'])) {
    if($_SESSION['logon'] == 0)
      header("location:index.php");
  }
?>

<head>
  <title>Home Page For Project Site [Project Name]</title >
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/3.3.7/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>

<?php include "header.php"; ?>

<body>
<div class="container">
  <div class="row">
    <div class="col-md-6 col-md-offset-3">
      <div class="panel panel-login">

```



```
<div class="panel-heading">
  <div class="row">
    <div class="col-xs-6">
      <a class='btn btn-primary' href='admin_home.php'>back</a>
      <br>
      <br>
    </div>
  </div>
  <hr>
</div>
<div class="panel-body">
  <div class="row">
    <div class="col-lg-12">
      <form id="add_lift" name="add_lift"
      action="validate_add_lift.php" method="post" role="form"
      style="display: block;">
        <div class="form-group">
          <input type="text" name="lifts_to_add" id="lifts_to_add"
          tabindex="1" class="form-control"
          placeholder="Number of Lifts to Add" value="">
        </div>
        <div class="form-group">
          <div class="row">
            <div class="col-sm-6 col-sm-offset-3">
              <input type="submit" name="submit" id="login-submit"
              tabindex="4" class="form-control btn btn-primary"
              value="Submit">
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>
```

```
</div>
</div>
</div>
</body>
</html>
```

**Listing 16:** Remove Item from Queue

```
<?php

    //goal: to remove entry from queue because it's complete or
    //cancelled

    include "db_config.php";

    //connect to mysql server
    $conn = new mysqli($db_host, $username, $password, $dbname)
        or die("Error" . mysqli_error($conn));

    //get company name
    if(isset($_SESSION['email'])) {
        $query = "SELECT company FROM $login_info WHERE email = '"
            . $_SESSION['email'] . "'";
        $result = mysqli_query($conn, $query);
    }
    while($row = mysqli_fetch_array($result)) {
        $company = $row['company'];
    }

    //get lift num from url
    if(isset($_SESSION['num']) && isset($_SESSION['order'])) {

        $query = "UPDATE $company SET qfn".$_SESSION['order'].
            "= NULL, qtask".$_SESSION['order']. "= NULL WHERE lift_num = '"
            . $_SESSION['num'] . "'";
        $result = mysqli_query($conn, $query);
    }
}
```

```

    }

    if (isset($_SESSION['num'])) {
        header("location: /~mrubia/TrackMyHoist/view_queue.php?num="
            . $_SESSION['num']);
    }
?>

```

**Listing 17:** Request Lift Page

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN/"
<html xmlns="http://www.w3.org/1999/xhtml">

<!-- request lift page
    request lift number, what floor you need it on, and task (optional)
-->

<?php
    if (!session_id()) session_start();
    if (isset($_SESSION['logon'])) {
        if ($_SESSION['logon'] == 0) {
            header("location: index.php");
            die();
        }
    }
?>

<head>
    <title>Home Page For Project Site [Project Name]</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>

<?php include "header.php"; ?>

```

```
<body>
<div class="container">
  <div class="row">
    <div class="col-md-6 col-md-offset-3">
      <div class="panel panel-login">
        <div class="panel-heading">
          <div class="row">
            <div class="col-xs-6">
              <!-- check if admin or employee -->
              <?php
                // if admin
                if (isset($_SESSION['logon'])) {
                  if ($_SESSION['logon'] == 1) {
                    echo "
                      <a class='btn btn-primary'
                        href='admin_home.php'>back</a>
                    ";
                  }
                }
                else if ($_SESSION['logon'] == 2) {
                  echo "
                      <a class='btn btn-primary'
                        href='home.php'>back</a>
                    ";
                }
              }
            </div>
            <br>
            <br>
          </div>
        </div>
        <hr>
      </div>
      <div class="panel-body">
        <div class="row">
          <div class="col-lg-12">
```

```
<form id="request" name="request"
action="validate_request_lift.php" method="post" role="form"
style="display: block;">
  <div class="form-group">
    <input type="text" name="lift_num" id="lift_num" tabindex="1"
      class="form-control" placeholder="Lift Number*" value="">
  </div>
  <div class="form-group">
    <input type="text" name="floor" id="floor" tabindex="1"
      class="form-control" placeholder="Floor*" value="">
  </div>
  <div class="form-group">
    <input type="text" name="task" id="task" tabindex="1"
      class="form-control" placeholder="Task" value="">
  </div>
  <div class="form-group">
    <div class="row">
      <div class="col-lg-12">
        <span style="color:gray; font-size: .8em;">
          <p>*Required Fields</p>
        </span>
      </div>
    </div>
  </div>
  <div class="form-group">
    <div class="row">
      <div class="col-sm-6 col-sm-offset-3">
        <input type="submit" name="submit" id="login-submit"
          tabindex="4" class="form-control btn btn-primary"
          value="Submit">
      </div>
    </div>
  </div>
</form>
</div>
```

```
    </div>
  </div>
</div>
</div>
</div>
</div>

</body>
</html>
```

**Listing 18:** TCP Connection Information

```
<?php

    $host = "10.0.0.12";
    $port = 10001;
    $message = "Hello ";
    set_time_limit(0);
?>
```

**Listing 19:** Results Page

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN//"  
<html xmlns="http://www.w3.org/1999/xhtml">

<?php session_start(); ?>

<head>
    <title>Home Page For Project Site [Project Name]</title >
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/
3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>

<?php include "header.php"; ?>

<body>
```

```

<h1 class="text-primary text-center" style="margin-top: 5em">
  Successfully Submitted Request!</h1>
<h3 class="text-primary text-center">Added to queue</h3>
</body>

```

**Listing 20:** Registration Results Page

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN/"
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <meta http-equiv="Content-Type" content="text/html;
  charset=utf-8" />
  <link rel="stylesheet"
  href="http://www.w3schools.com/lib/w3.css">
  <link rel="stylesheet"
  href="https://maxcdn.bootstrapcdn.com/bootstrap/
  3.3.7/css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="styles.css">

</head>

<?php include "header.php"; ?>

<body>

  <h1 class="text-primary text-center" style="margin-top: 5em">
    Successful Registration!</h1>
  <h3 class="text-primary text-center">Proceed to login page</h3>

</body>
</html>

```

**Listing 21:** Lift Request Results Page

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN/"
<html xmlns="http://www.w3.org/1999/xhtml">

```

```
<?php session_start (); ?>

<head>
    <title >Home Page For Project Site [Project Name]</title >
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/
bootstrap/3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>

<?php include "header.php"; ?>

<body>
    <h1 class="text-primary text-center" style="margin-top: 5em">
    Successfully Added Lift!</h1>
    <h3 class="text-primary text-center">
    Viewable on home page</h3>
</body>
</html>
```

**Listing 22:** Test Page

```
<?php
    $floor = 23;
    $url=$_SERVER['REQUEST_URI'];
    header("Refresh: 1; URL=$url");
?>
```

**Listing 23:** Validation for Lift Registration

```
<?php

    session_start ();

    ob_start ();

    if (isset ($_POST [ 'submit ' ]))
    {
        include "db_config.php";
```



```
$result = 0;

//connect to mysql server
$conn = new mysqli($db_host, $username, $password, $dbname)
        or die("Error" . mysqli_error($conn));

//get number of lifts to add and convert to int
$lifts_to_add = (int)$_POST['lifts_to_add'];

//get company name
if (isset($_SESSION['email'])) {
    $query = "SELECT company FROM $login_info WHERE email =
        '$_SESSION['email']'";
    $result = mysqli_query($conn, $query);
}

//get company name from query
while($row = mysqli_fetch_array($result)) {
    $company = $row['company'];
}

//get all information from company's lift lists
$query = "SELECT * FROM $company";
$result = mysqli_query($conn, $query);
$num_lifts = mysqli_num_rows($result);
//$num_lifts += 1;

$j = $num_lifts + 1;
$end = $lifts_to_add + $num_lifts;
//using for loop
//add number of lifts to table starting from last number in
for($i = 1; $i <= $lifts_to_add; $i++) {
    $query = "INSERT INTO $company (lift_num) VALUES ('$j')";
    $result = mysqli_query($conn, $query);
}
```

```
                $j++;
            }

        }
header("location: successlift.php");
ob_end_flush();

?>
```

**Listing 24:** Validate Login Information

```
<?php

    //start session
    session_start();

    ob_start();

    if (isset($_POST['submit']))
    {

        include "db_config.php";

        //connect to mysql server
        $conn = new mysqli($db_host, $username, $password, $dbname)
            or die("Error" . mysqli_error($conn));

        //database fields:
        //user email, password, admin (acts as bool)
        //can set flag for admin to set admin permissions

        //validation: array for error messages 'errors[]'
        //error flag: isError == true if errors
        //password var: psw
```

```
//admin in db is 1 char: y (yes) or n (no)
//if y, then user is an admin

$email = $_POST['email'];
$psw = $_POST['password'];
//protect from mysql injection
$email = stripslashes($email);
$psw = stripslashes($psw);
$email = mysqli_real_escape_string($conn, $email);
$psw = mysqli_real_escape_string($conn, $psw);

//session var to check if user is logged in
//0 is not logged in, 1 is admin user, and 2 is regular employee
$_SESSION['logon'] = 0;

//for queue information later
$_SESSION['num'] = 0;
$_SESSION['order'] = 0;

//array for error validations
$errors = array();
//error flag validation
$isError = false;
$errorMsg = '';

//no email entered
if($email == '')
{
    $errors['email'] = 'Enter Email';
    $isError = true;
}

//no password entered
if($psw == '')
{
```

```
        $errors['psw'] = 'Enter Password';
        $isError = true;
    }

    if($isError)
    {
        foreach($errors as $error) {
            $errmsg .= $error . '<br />';
        }
        //$_SESSION['ERRORS'] = $errors;
        //session_write_close();
        //header("location: login.php");
        //exit();
    }
    $check = 0;

    $query = "SELECT * FROM $login_info WHERE email = '$email' AND
    password = '$psw'";

    // if admin is logging in

    $result = mysqli_query($conn,$query);
    // login success
    if(mysqli_num_rows($result) == 1)
    {
        $_SESSION['email'] = $email;
        $row = mysqli_fetch_array($result);
        //if admin, redirect to admin home page
        //register global var admin
        //session_register("admin");
        //header("location:admin_home.php");
        $_SESSION['logon'] = 1;
        $check = 1;
        header("location:admin_home.php");
        //else redirect to employee home page
```

```
}
else
{
    // if employee is logging in
    $query = "SELECT * FROM $login_info WHERE email = '$email'
AND sub_password = '$psw'";
    $result = mysqli_query($conn,$query);

    // login success
    if (mysqli_num_rows($result) == 1)
    {
        $_SESSION['email'] = $email;
        $_SESSION['logon'] = 2;
        $check = 1;
        header("location:home.php");
    }
}
if ($check == 0)

{
    //redirect back to login page
    header("location: login.php");
    exit ();
}
}
ob_end_flush ();
?>
```

**Listing 25:** Validate Lift Request Page

&lt;?php

```
/*
validate requests
add to queue
search for $company_queue table
```

```
add request to table

currently queue has max number of 8 slots for requests, because
it's hardcoded

table rows will be numbered: task1, task 2, ... , taskn
search rows of that number to see if any spots are empty (null)
if null
    add task to that row
else
    create new column in that row with incremented task name
    (task_n+1)
*/
session_start();

ob_start();

if(isset($_POST['submit']))
{
    include "db_config.php";

    //connect to mysql server
    $conn = new mysqli($db_host, $username, $password, $dbname)
        or die("Error" . mysqli_error($conn));

    $lift_num = $_POST['lift_num'];
    $floor = $_POST['floor'];
    $task = $_POST['task'];

    $lift_num = stripslashes($lift_num);
    $floor = stripslashes($floor);
    $task = stripslashes($task);
    $lift_num = mysqli_real_escape_string($conn, $lift_num);
    $floor = mysqli_real_escape_string($conn, $floor);
    $task = mysqli_real_escape_string($conn, $task);
```

```
//get company name
if(isset($_SESSION['email'])) {
    $query = "SELECT company FROM $login_info WHERE email = '"
        .$_SESSION['email']."'";
    $result = mysqli_query($conn,$query);
}
while($row = mysqli_fetch_array($result)) {
    $company = $row['company'];
}

$query = "SELECT * FROM $company WHERE lift_num = '"
    .$lift_num."'";
$result = mysqli_query($conn,$query);

//create vars for first slots of info in queue
while($row = mysqli_fetch_array($result)) {
    $qfn1 = $row['qfn1'];           //floor num
    $qtask1 = $row['qtask1'];       //task
    $qfn2 = $row['qfn2'];
    $qtask2 = $row['qtask2'];
    $qfn3 = $row['qfn3'];
    $qtask3 = $row['qtask3'];
    $qfn4 = $row['qfn4'];
    $qtask4 = $row['qtask4'];
    $qfn5 = $row['qfn5'];
    $qtask5 = $row['qtask5'];
    $qfn6 = $row['qfn6'];
    $qtask6 = $row['qtask6'];
    $qfn7 = $row['qfn7'];
    $qtask7 = $row['qtask7'];
    $qfn8 = $row['qfn8'];
    $qtask8 = $row['qtask8'];
}
}
```

```

    $i = 1;

    for($i = 1; $i < 9; $i++) {
        //search for empty slot in queue
        if("${qfn" . $i} == NULL) {
            $query = "UPDATE $company SET qfn".$i."=' $floor ', qtask".
                $i."=' $task ' WHERE lift_num = $lift_num";
            $result = mysqli_query($conn,$query);
            header("location:success.php");
            break;
        }
    }
}

header("location:success.php");
ob_end_flush();

?>

```

**Listing 26:** View the Lift Queue

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN/"
<html xmlns="http://www.w3.org/1999/xhtml">

<!-- check if user is logged in to access page -->
<?php
    if(!session_id()) session_start();
    if(!isset($_SESSION['logon'])) {
        header("location:~/mrubia/TrackMyHoist/index.php");
    }
?>

<head>
    <title >View Queue</title >
    <link rel="stylesheet"
    href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/

```



```
bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="styles.css">
<link
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.4.0/
css/font-awesome.min.css"
rel='stylesheet' type='text/css'>
</head>

<script>
$(document).ready(function() {
    $('[data-toggle=offcanvas]').click(function() {
        $('.row-offcanvas').toggleClass('active');
    });
});
</script>

<?php include "header.php"; ?>

<body>

<div class="col-md-10 col-md-offset-1">
    <div class="panel panel-default panel-table">
        <div class="panel-heading">
            <div class="row">
                <div class="col col-xs-6">

<?php
    if(isset($_SESSION['logon'])) {
        if($_SESSION['logon'] == 1) {
            echo "
                <a id='back-btn' class='btn btn-primary' href='/'
                ~mrubia/TrackMyHoist/admin_home.php'>back</a>
            ";
        }
        if($_SESSION['logon'] == 2) {
            echo "
```

```

                                <a id='back-btn' class='btn btn-primary' href='/'
                                ~mrubia/TrackMyHoist/home.php'>back</a>
                                ";
                                }
                                } ?>

<hr>

<?php

    //get lift num from url
    if(isset($_SESSION['num']) && isset($_GET['num'])) {
        $_SESSION['num'] = $_GET['num'];
    }
    echo "
    <h3 class='panel-title '>Lift Queue</h3>
</div></div></div>
    ";
    include "db_config.php";

    //connect to mysql server
    $conn = new mysqli($db_host, $username, $password, $dbname)
        or die("Error" . mysqli_error($conn));

    //get company name
    if(isset($_SESSION['email'])) {
        $query = "SELECT company FROM $login_info WHERE email = '"
        .$_SESSION['email']."'";
        $result = mysqli_query($conn,$query);
    }

    while($row = mysqli_fetch_array($result)) {
        $company = $row['company'];
    }

```

```
if (isset($_SESSION['num'])) {

    $query = "SELECT * FROM $company WHERE lift_num = '"
    .$_SESSION['num']. "'";
    $result = mysqli_query($conn,$query);

    while($row = mysqli_fetch_array($result)) {
        $qfn1 = $row['qfn1'];
        $qtask1 = $row['qtask1'];
        $qfn2 = $row['qfn2'];
        $qtask2 = $row['qtask2'];
        $qfn3 = $row['qfn3'];
        $qtask3 = $row['qtask3'];
        $qfn4 = $row['qfn4'];
        $qtask4 = $row['qtask4'];
        $qfn5 = $row['qfn5'];
        $qtask5 = $row['qtask5'];
        $qfn6 = $row['qfn6'];
        $qtask6 = $row['qtask6'];
        $qfn7 = $row['qfn7'];
        $qtask7 = $row['qtask7'];
        $qfn8 = $row['qfn8'];
        $qtask8 = $row['qtask8'];
    }

    echo "
    <div class='panel-body'>
        <table class='table table-striped table-bordered
        table-list'>
            <thead>
                <tr>
                    <th><em class='fa fa-cog'></em></th>
                    <th>Floor Requested</th>
                    <th>Task</th>
                </tr>
```

```
        </thead>
        <tbody>
        <tr>
";
for($i = 1; $i < 9; $i++) {
    if("${qfn" . $i} == NULL) {
        continue;
    }
    echo "
        <td>
            <td>".${qfn" . $i}." </td>
            <td>".${qtask" . $i}." </td>
        </td>
";
}

echo "
    </tr>
</tbody>
</table>";
}

?>

<!-- if admin, add ability to edit queue -->

<?php
    if(isset($_SESSION['logon'])) {
        if($_SESSION['logon'] == 1) {
            echo "
                <a class='btn btn-primary' href='edit_queue.php?num="
                .$_SESSION['num'].' '>Edit Queue</a>
            ";
        }
    }
}
```

?>

</body>

</html>