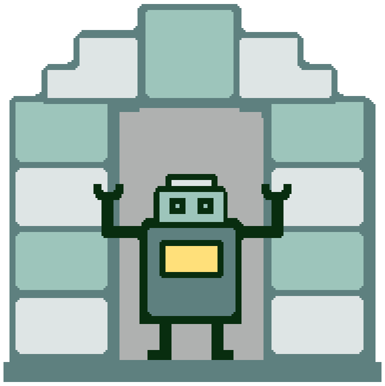
Requirements Document

12/4/2018

Version 1.2



Keystone Robotics
Robot-Assisted Tours

Project Sponsor:

Michael Leverington

Faculty Mentors:

Austin Sanders, Jun Rao

Team Members:

Hailey Ginther, Shannon Washburn, Gabrielle Halopka, Falon Ortega

Contents of this document accepted as baseline requirements for the project:

Client Signature: ____ Date:__________________________

Team Signature: ____ Date:__________________________

Table of Contents

1.0 - Introduction

Northern Arizona University's Engineering building is the most important stop of campus tours for future engineering students. The labs, project rooms, and lecture halls of the building are where they will be studying for the next 4+ years. This means the impression touring groups get of the facilities during their time on the tour is an essential contributing factor to attracting and retaining new students. Attracting the attention of these students with physical evidence of the work accomplished by seniors in the department will help convince these new individuals to enroll.

The ultimate purpose of this project is the creation of a robot capable of autonomously giving tours of the engineering building, fulfilling this need for a captivating introduction to the projects NAU students can accomplish thanks to their coursework. Our goal as the starting force behind this project is to put what we have identified as the basic level of hardware and software this robot needs into practice.

1.1 - Keystone Robotics' Goal

For the scope of this capstone project, the client's goal is to have an assembled robot that can be moved from user inputs. Our other major goals during this project are to perform tests to ensure the safety and stability of hardware and to leave behind detailed documentation for future students' use during follow-up projects. The basic system our team has outlined to solve our problems involves using microcontrollers that send signals to motors through motor controller circuits. The design must meet certain requirements as delineated by our client, such as the ability to handle its own weight of at least 100 pounds and be driven by a high-voltage battery(s). Additionally, this robot must be able to construct a map of the engineering building and be able to successfully navigate at least one floor with user input, with the stretch goal of the project being autonomous navigation.

1.2 - Client's Vision

Our client, Dr. Michael Leverington, is a professor at Northern Arizona University's School of Informatics and Computing. The professor has a Ph.D. in Education, Masters of Computer Science and Psychology, and a Bachelors in Physics. Additionally, he is an avid follower of developments in robotics and is eager to bring the multidisciplinary topic to his department. He has tasked our team with the initial planning, assembly, and programming of a robot potentially capable of giving tours of NAU's engineering building with a level of autonomy. Our client sees the robot as a solution to two main problems-

1. The need for the automation of tours. The issues with the current tour workflow include:
    a. Faculty time - In cases where professors give tours, the tours will be accurate and informative of the type of work done in the building, but will take up valuable time that may detract from more urgent academic work
    b. Lack of consistency - The department has no established tour guidelines, leaving each tour leader to disseminate information at their own discretion, leaving room for potential gaps or discrepancies in the coverage of building information
    c. Potential for lack of expertise - In cases where tours are given by non-engineering faculty, there is a chance the tour leader lacks sufficient knowledge of the building and the department as a whole, leading to significant gaps in the tour's coverage of topics

2. The need for a robotic framework that future student teams could use as a foundation for other projects. Currently, the engineering building has no robotic framework available to students, leading to the following issues:
    a. Price of pre-made robots - Fully fabricated robots with the desired specs for high-demand projects are outside of the average student's price range, and may not have the full suite of features needed for a specific project
    b. Hardware barriers - Building a robotics project is a multidisciplinary challenge, meaning a team comprised of students only from one major, such as computer science or electrical engineering, would be unable to attempt many robotics endeavors without taking the time to learn concepts outside the scope of their desired projects

As the first team to work on this project, our focus is entirely on finding the solution to the second of the two problems. However, the first remains relevant in our process, since we must design and build the robot with the end goal of tours in mind. For more detailed information on the problems the Keystone Robotics team is facing, refer to section 2 of this document.

2.0 - Problem Statement

The problem our client has presented to us is that there is currently no robotics project or simple way to build robots for CS students. Our client's desire is that our tour giving robot will serve as a blueprint for future projects. The overarching goal of the project, or the intended use of the robot, is that it is mostly autonomous and able to give tours of the engineering building. For our purposes, this is considered to be a stretch goal that focuses the trajectory of our project.

The client has a passion for robotics and would like to see more robotics projects in the department as there is a lack of robotics projects for CS students and of the research robots that are available, many are very expensive and difficult to modify. One of our tasks is to establish a framework for more robotics projects for current and future students. While our particular robot is made to give tours, this project should be easy to modify and should have sufficient resources for students to follow and understand. For example, if a student wanted to design a robot that picks up trash instead of giving tours, this project should provide them with a resources to do so. The parts of the robot should also be inexpensive enough so that the department or a student with limited resources can afford to add components or make design changes.

For the robot itself, it is intended to autonomously give tours of the engineering building. Currently, NAU's True Blue ambassadors are in control of orientation and giving tours of the campus, including the engineering building. While they may be able to provide basic information, they may not be engineers. The True Blue ambassadors may not be knowledgeable enough about the engineering building or department as a whole to give adequate information. Some professors are willing to give tours, but these tours could take up valuable time from their academic schedules. With the robot, a guided tour will always be available. The tour robot will simplify the guided tour process and allow staff to spend their time more effectively. The project should also be able to catch the attention of not only current students but incoming students as well to attract them to potential robotics projects.

3.0 - Solution Vision

For the tour giving robot, the solution will be a combination of hardware and software components. The hardware component will be the physical robot, such as the barrel, motors, wheels, and frame. The focus of the hardware team is to build the physical robot and make the robot move with assistance. A simple diagram of the model is pictured below.

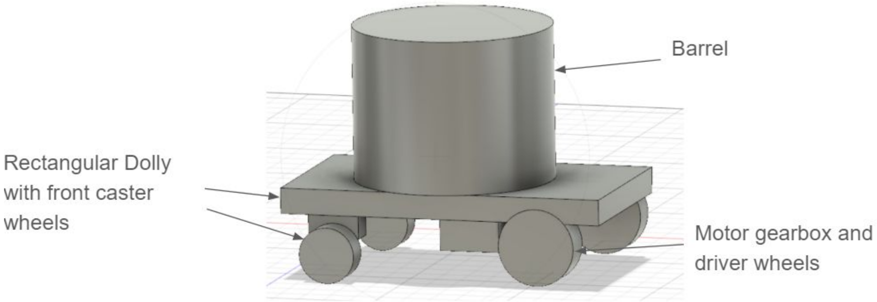


Figure 1. Simplified system model.

The software portion of the solution focuses on how the robot reacts to an input, more specifically, how it moves in an environment. This involves software that allows navigation and mapping. The software should be able to take input and make decisions for the robot based on that input. The software will use a series of sensors in order to collect the necessary data and connect to the motors in order to control them. The software will also be used for any sort of user interface or commands. A diagram of the architecture is shown below.

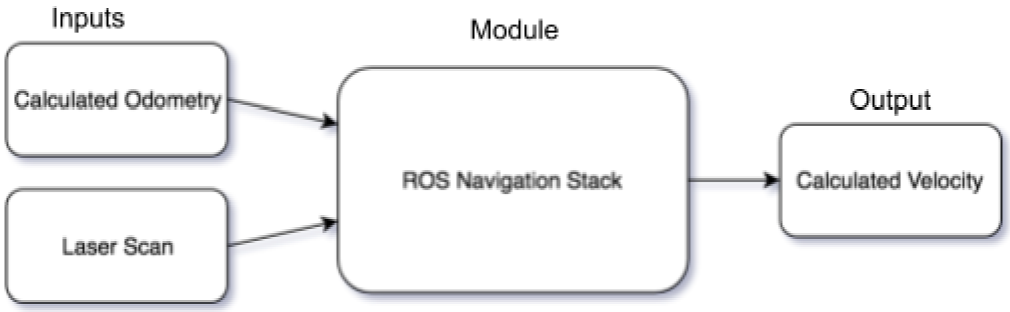


Figure 2. Flowchart of the software process.

The other aspect of the solution is the ability for another team to follow the project to create their own robot. To make this more achievable, the team plans to develop a manual for new users to understand how the current robot works and how it can be modified. Along with the manual, there will be documentation for any incoming teams to follow along and better understand what decisions were made and why. This ensures that if there any gaps in the manual, the documentation of the process should fill in the holes.

The pieces should be easily modified or interchangeable. All components of the robot should be robust, or able to still operate or notify the user of an issue if a component has been removed or modified in such a way that it cannot operate. Parts should be able to be swapped out with little to no issue or expense.

4.0 - Project Requirements

In order to understand how to best form our solution and solve our problem, we formulated a series of requirements in order to quantifiably know when our project has been completed.

4.1 - Introduction

In order to obtain the requirements for our project, we started by conducting meetings with our client in order to establish his vision and expectations for our team. This provided us with the context to conduct our own technical research to determine what would be feasible for our team to complete within the scope of a two-semester project. As we gained more insight into our problem and solution, we continued to meet with our client to present him with our research and refine our requirements. This method resulted in our four domain level requirements:

1. The robot will be capable of basic navigation
2. The robot will be expandable to future projects
3. The robot will operate safely
4. The robot will be usable for a technical end user

4.2 - Functional Requirements

Our functional requirements are composed of all of the functions that our robot will be able to perform. They are laid out in a hierarchical manner with more general requirements at higher levels and more detailed requirements which fall under those at lower levels. We will implement the following functionalities for our robot to perform:

1. The robot will be able to navigate a path between two points on a single floor
    - 1.1. The robot will be able to accurately calculate odometry data
        - 1.1.1. The robot will be able to acquire and use wheel encoder data
            - Wheel encoder data from each drive motor will provide us with information on how many times the wheels have turned as well as the direction of the rotation. This data along with our knowledge of the circumference of the wheels will allow us to calculate how far the robot has traveled. The relationship between the two wheel's encoder data will allow us to also calculate the robot's velocity based on the differing velocities of the left and right

motors. These sensors will be connected to the Arduino where the data will be read, processed, and transformed into a ROS (Robot Operating System) message containing the approximate odometry.

1.1.2. The robot will be able to acquire and use inertial measurement unit (IMU) data

- IMU's provide a combination of accelerometer and gyroscope data which will allow us to know the speed and rotation of the robot at any given time. This sensor will be connected directly to the Arduino where we will process the data and transform it into a ROS message.

1.1.3. The robot will be able to combine wheel encoder and IMU sensor readings

- Due to the inaccurate tendencies of wheel encoder data because of potential wheel slippage, it needs to be combined with IMU sensor data in order to get a more accurate reading. ROS provides a package called *robot_pose_ekf* which can take in wheel encoder, IMU, and visual odometry messages and combines them into an approximate 3D pose for the robot and then sends a ROS message containing the pose.

1.2. The robot will be able to create a 2D map of its environment

1.2.1. The robot will be able to acquire and use laser scan data

- ROS's navigation stack requires a laser scanner source in order to perform mapping, localization, and navigation functions. Using the *libfreenect* and *pointcloud_to_laserscan* packages we can connect a Microsoft Kinect sensor to our Raspberry Pi using a power adapter and convert the resulting point-cloud data into a pseudo laser scan. The data is published to the *sensor_msgs/LaserScan* topic and used by the navigation stack.

1.3. The robot will be able to localize itself within a 2D map

- Given a 2D map previously created, the ROS navigation stack comes up with multiple estimated positions based on the current readings coming from the laser scanner. The localization process involves the robot moving, likely rotating around its current position, and comparing laser scan data to the data created in the mapping process. This updates the estimates as the robot learns more about its surroundings and eventually they converge on the correct location and direction within the map.

1.3.2. The user will be able to send robot approximate location and heading

- Given a 2D map which the robot has created, ROS provides functionality which allows a user to view the map using *rviz*. *Rviz* is a tool which gives a visual representation of sensor and map data. It also allows users to select an approximate location and heading on the map for the robot's current position. This allows for faster and more accurate localization.

1.4. The robot will be able to send velocity data to control two motors independently

- The ROS navigation stack publishes a *geometry_msgs/Twist* message containing the x, y, and angular velocities which need to be interpreted and transformed into commands which can be sent to the motors directly via the Arduino.

2. The robot will be able to avoid hazards
    - 2.1 The robot will be able to reroute around obstacles
        - 2.1.1. The robot will be able to detect obstacles
        - 2.1.2. The robot will be able to change path when encountering obstacles
        - 2.1.3. The robot will be able to warn people and obstacles of its presence
    - 2.2. The robot will be able to reroute to avoid drop-offs
        - 2.2.1. The robot will be able to detect drop-offs
        - 2.2.2. The robot will be able to change route when encountering drop-offs
3. The robot will be manually controllable
    - 3.1. The robot will be able to interpret joystick commands
    - 3.2. The robot will be able to use interpreted joystick commands to signal the motors
4. The robot will have an emergency stop mechanism which will disconnect the motors from the power supply.

4.3 - Non-functional Requirements

Our non-functional requirements comprise all of our hardware performance specifications and safety requirements as well as our requirements for expandability and usability. They are laid out in a hierarchical manner similar to our functional requirements.

1. Robot's velocity will be at least 1.5 m/s while moving at least 100 lbs.
   1.1. Robot's motors will be able to provide a torque of at least 441 Nm to move one meter
   1.2. Robot's wheels will be able to support the weight of the robot without breaking under pressure
2. Robot will be able to operate continuously for at least 2 hours
   2.1. Robot needs 20 watts for the motors
      2.1.1. Robot's battery system will be at least 20 watts.
3. Robot will not pose a risk to its environment or itself
   3.1. Under normal operating conditions, the components will not exceed 185° Fahrenheit
   3.2. Robot will be able to cope with minor to moderate impacts
      3.2.1. A minor impact is defined as light bumping or jostling, such as running over an uneven surface that may slightly tilt the robot
      3.2.2. Moderate is anything that would stop the robot, such as hitting a wall
      3.2.3. Coping is considered to mean the robot will endure no more than minor bumps or scrapes that do not affect the functionality of the robot.
   3.3. Robot housing and attachments will be secure
      3.3.1. Outer components will not fall off after navigating one floor of the Engineering building.
      3.3.2. Inner components will not shift after navigating one floor of the Engineering building.
   3.4. Robot will have emergency stop mechanism which will disconnect power to motors within 1 second
4. Robot will be expandable to future projects
   4.1. Robot will provide space to add additional hardware components
   4.2. Robot software will be readable and modifiable as per the client's coding style requirements document
   4.3. Robot hardware will be separated into two modules: microcontrollers and motors. Each module will be modifiable.
5. Robot will be usable to a technical end user

5.1. Robot will have a comprehensive manual of operation
    5.1.1. Each part of the design will be extensively documented
5.2. Input, output and power ports will be accessible

4.4 - Environmental Requirements

When formulating our functional and nonfunctional requirements, we also needed to take into consideration any constraints that would affect the solution to our problem. During the process of eliciting our requirements during client meetings we also established the following four constraints for our project:

1. Budget: Parts must be cost-effective
2. Scale: Client envisions a large-scale robot
3. Housing: Components must be housed in a 30-gallon barrel
4. Location: Robot only needs to navigate indoors

4.4.1 - Budget

Our budget is one of our biggest constraints and affects our design by limiting the hardware components we can choose. Our client has given us a soft-cap budget of $500 dollars for all hardware components including motors, batteries, mounting hardware, adapters, micro-controllers, and sensors. He has indicated that this limit is, however, negotiable and if presented with necessary components that may go over our budget he is willing to be flexible. This does not change the fact that we need to do the necessary research to find hardware components that are viable for our project at the most reasonable price possible.

4.4.2 - Scale

Due to our client's desire for a large-scale and robust robot, we have listed this as one of our constraints. The large size of our robot also limits the hardware components we can consider due to the need for the higher amount of power required to make it move. This constraint conflicts with our budget constraint as the larger the size of the robot, the more expensive components will be required. Keeping a balance between compromising on either size or budget has been one of our team's biggest challenges thus far.

4.4.3 - Housing

Another requirement set by our client is that our hardware components be housed in a 30-gallon barrel. This has both guided and limited our overall design decisions regarding construction and sensor placement.

4.4.4 - Location

Because our robot is only expected to function within Northern Arizona University's Engineering building, we will not have to take into account outdoor navigation. This guided our decisions involving the wheels for our robot as well as positional sensors and software packages.

5.0 - Potential Risks

As with most projects involving autonomous or higher-power devices, there are risks associated with the development of this robot. The consequences of errors in this project have the chance of resulting in damage to the robot, the robot's surroundings, and harm to people around it. Since this project is multi-disciplinary, the team decided to separate the possible risks into two categories, hardware, and software. There were three parameters used to evaluate each of the risks in this project: likelihood, concern level, and mitigation strategy.

The likelihood category is rated on a scale from one to five, where one is low and five is high. Since likelihood refers to the probability that risk will occur, a one means that the risk is very unlikely, while a five is very likely, or almost certain. The concern level is quantized into three levels: low, mid, and high. Concern level is the amount of attention the team feels they need to pay attention to this risk; in other words, it is the level the risk could damage the project and its outcome if it does happen. The final section is the mitigation strategies for each risk. Mitigation strategies have been devised by the team members for each risk in order to decrease their chances of happening as much as possible. Many risks have multiple strategies to combat the problems from multiple angles. The method used to devise these strategies was based on the consideration of the following:

- Hardware:
    - Can the risk be lessened by selection of a more reliable part?
    - Can the risk be lessened with the addition of extra parts?
    - Can the risk be lessened via some alteration of the assembly design?
- Software:
    - Can safety measures be added within the code that may mitigate that risk?
    - Can tests be performed on the software to find errors that may cause that risk to occur?
    - Can an addition, hardware or software based, provide some sort of safety measure to halt the robot in the event of a risk occurring?

In summary: Each item identified in the following two subsections is described, then the consequences should the risk occur are outlined. Finally, each risk's likelihood is calculated using the 1-5 scale described in the previous paragraph. A table of the information presented in this section can be found in the appendix at the end of this document.

5.1 - Hardware

For a summary outlining the hardware risks, see table 1 in the appendix.

- Overheating
    - Description: Due to the nature of the requirements for this system, the motors and batteries chosen have to be high power. Overheating is a concern for any circuit that is designed poorly or without the proper heat dissipation components. High wattage parts must be selected to mitigate this risk.
    - Consequences: Overheating can lead to melting of components or damaging necessary circuitry for the device to function. Damage to circuits due to overheating is usually irreversible, and can only be solved by purchasing new replacement parts. If the circuitry does not have the proper heat mitigation methods used, then the parts used will have to be replaced so often that it will greatly inconvenience the client.
    - Likelihood: 3 / 5 - There will be an unavoidably high amount of heat emitted from this device and its circuitry.
    - Mitigation strategy - Dealing with it properly should be fairly simple. Holes may be cut into the base to reroute the heat, and different forms of heat sinks may be used, ideally copper finned.
- Combustion
    - Description: AGM (Absorbent Glass Mat) Lead-Acid batteries can possibly combust, especially if not certified. Lead-Acid batteries are commercially common, are in many types of vehicles and robots, from airplanes to kiddie cars. The chance of combustion is typically very small, especially for the smaller models available. The biggest causes of combustion would be extreme temperatures and/or overcharging the batteries, or very hard impacts.
    - Consequences: If the battery explodes, it will release debris in about a five-foot radius. The battery will likely catch fire, depending upon the internal state of the battery at the time of combustion. Then, the robot itself may catch on fire, ruining all of the components inside.
    - Likelihood: 1 / 5 - The batteries we are buying are certified, so any separator issues or other manufacturer defects are very low. The battery is also unlikely to catch fire if it explodes, so the damage itself would probably be contained to just the battery. The likelihood of severe, high force, impacts is also very low, because it won’t ever leave the engineering building, so usually it will just be students running into the device as opposed to another vehicle.
    - Mitigation Strategy: The circuit we are designing will use multiple methods to decrease the heat of the overall system, so the battery should not be exposed to any high temperatures at all. The system and software design for the robot also

take into account the multiple chances for collision, and any hard collisions with this robot are very unlikely. This system will be using a low amperage, smart charger that can test the charge of the battery and control the amount of amperage from zero to its full charging state.

- Loosening/Detaching Components
    - Description: This system has multiple components, including the microcontrollers, motors, and motor driver. If any of these components were to detach from the device, there is a good chance the robot will malfunction. These detachments can come from poor stabilization, soldering, and storage.
    - Consequences: Depending on the components that are detached, the robot could function incorrectly or completely stop working. If the connection between the battery and the motor driver are compromised, all power to the wheel is cut off. If any of the motors are disconnected, the robots' movement will become compromised. If any of the wheels were to come loose and fall off, the robot could fall over, and the motors could become damaged if the spinning shaft touches the floor. Finally, if the microcontroller or motor driver circuits become disconnected, then control to the device shuts off and it will move completely unpredictably and possibly dangerously, or not at all.
    - Likelihood: 1 / 5 - Custom motor mounts are going to be ordered for the motors, so they should remain very secure. The microcontrollers and motor drivers are to be stored in their own compartments in the barrel or next to the wheel depending on any issues the device has with heat dissipation. The circuit will be soldered securely, with heat sinks attached once the hardware team is sure the circuit functions correctly, so no wired connection should become undone. After all of the components are attached, the robot will undergo rigorous testing that covers and overcompensates for any possible use cases to make sure the connections are all stable.
    - Mitigation Strategy: The circuit will use careful and robust soldering techniques, as well as high-grade heat sinks to secure the circuit in place. Inside the barrel, our client has imagined a shelving system that can store the batteries, microcontrollers, and motor drivers that can be lifted out for maintenance and update purposes. This shelf should be able to secure most of the components because it will be made by a very competent, local welding and machine shop. This same shop is where the motor mount was made as well, which should keep the motors and device together.

- Battery Leakage
    - Description: AGM Lead-Acid batteries can only leak when the shell is physically broken. Afterwards, the battery may burn out, or combust. AGM lead acid batteries have an absorbent layer of fiberglass inside of them that absorbs the sulfuric acid inside if it leaks, so there should never be any acid coming out of the battery unless it is split open, or combusts from the inside.
    - Likelihood: 1 / 5 - The AGM Lead-Acid batteries that are being purchased for this project are new, and are not from a third party. The chances of leaking are very low, especially because they will be stored in a ventilated environment, with various heat mitigation techniques being used. The robot should never be in any situation where there would be hard impacts that could break it, such as from another vehicle, because of our client's intended use.
    - Mitigation Strategy: The battery will be bought new, and a "smart" charger will be bought for the battery as well so that the low amperage will not cause the battery to overheat, bulge, and leak, while still able to be charged overnight.

5.2 - Software

For a summary outlining the software risks, see table 2 in the appendix.

- Localization Errors
    - Description: Autonomous navigation for this project depends on comparisons of incoming sensor data to a pre-built 2-D map. If the robot is moved to an unmapped location or if the sensor data is not what is expected an error will occur.
    - Consequences: The robot has the potential to enter restricted areas where unexpected or dangerous obstacles may exist. Drop-offs, low curbs, and other dangers accounted for on the 2-D map would also be missing, increasing the robot's chance of falls and collisions.
    - Likelihood: 3 / 5 - The chances of the robot being booted in an unfamiliar location is dependant on the user, while losing track of its current position is dependant on the data returned from the sensors. The error has a low likelihood if users follow guidelines defined. Sensor data accuracy depends on the accuracy and placement of the devices and can be disrupted by blockages to fields of view, jostling from a passerby, and other unforeseen circumstances.
    - Mitigation Strategy: Include ability for a user to manually inform the robot of its approximate location on a 2D map. Also, in cases where the robot cannot find a route, program simple warning output of a sound or light to inform those nearby that it has lost its location and failed to reroute.

- Speed Calculation Error
    - Description: The speed of the robot is situational, dependant on whether the robot is moving forward, turning, reversing, or performing some combination of these movements. Calculations must be done for each of these movement procedures to generate values that are sent to the motors.
    - Consequences: Unexpected calculations sent to the motors can result in movement outside the upper and lower bounds of speed. This could lead to the robot moving much faster than intended, sudden stops, or stalling of the motors. These movements could lead to injury of objects and people nearby and could put excess stress on the motors.
    - Likelihood: 2 / 5 - The chances of this risk are calculated to be low on the basis that a comprehensive set of tests on the movements of the project would expose any situations where the speed falls outside of defined boundaries.
    - Mitigation Strategy: Formation of potential edge cases that could cause speed calculations must be done and then extensively tested to limit the chances.

- Loss of Control (Manual or Automatic)
    - Description: Physical disconnection of a wired controller or signal dropping between the robot and a connected computer could cause loss of manual control. Desynchronization and other software errors could cause loss of automatic control.
    - Consequences: The robot could continue on toward its last known direction, regardless of map data and sensor input. People, walls, and obstacles could be struck, and the robot and passerby could be harmed.
    - Likelihood: 2 / 5 - Loss of automatic control is expected to be of low likelihood since the software will be extensively tested in a controlled environment before deployment to public spaces. Manual control is also expected to be fairly low but slightly more likely than the automatic loss of control, due to the unpredictable nature of the external components like USB controllers and laptops.
    - Mitigation Strategy: Inclusion of a large and easily reachable emergency stop button that cuts power to the system immediately to halt all movement.

- Collision with Passerby or Obstacles
    - Description: Collisions may happen despite implemented safety measures. The robot could strike a person, wall, or another obstacle.
    - Consequences: People may be harmed by collisions from this robot since it could weigh as much as 100 pounds. Components of the robot may be harmed or shaken loose by these collisions as well. Property damage may also occur.

  - Likelihood: 4 / 5 - Despite preventative measures including testing and sensors, obstacles will almost certainly be hit. Walls and other known map data are less likely, but passerby has a strong chance of eventually being struck due to the fact that they move unpredictably.
  - Mitigation Strategy: First, we plan to add light or sound output devices to the robot that will activate to warn people nearby of its presence. Second, we plan to mount and test sensors to ensure their scope of vision covers the maximum amount of its immediate environment possible. Finally, we plan to test the framework and mounting of our hardware in a controlled environment to ensure our assembly remains stable after minor to moderate collisions.

- Falls/Failure to Detect Drop Offs
  - Description: In the case of the robot failing to notice drop-offs in a timely manner, the project could become stuck or fall.
  - Consequences: The robot falling could result in serious consequences, including damage of the robot, damage to surrounding objects, and harm of passersby. This robot is heavy and contains some sensitive parts, so the resulting damage from falling parts could render the robot inoperable and cause non-trivial injuries.
  - Likelihood: 2 / 5 - Stairs, small declines, and curbs will be incorporated into the map data with flags to avoid the areas surrounding them, reducing the likelihood of this risk. Sensors dedicated to recognizing drop-offs will further reduce the likelihood, resulting in our final estimation of ⅖.
  - Mitigation Strategy: Include warnings or stop points in 2D map around areas known to contain stairs, ledges, and other drop-offs. Place and program sensors on underside of robot that are capable of halting the robot the moment a ledge is seen.

6.0 - Project Plan

This project was broken into three phases: planning, testing, and assembly. The planning phase included team startup, planning, and documentation. This phase specifically involved becoming familiar with the client's problem, the requirements and goal of the project. Our team finalized these concepts in the Tech Feasibility and Requirements document.

While the planning phase was in progress, the team entered the testing phase. While working on describing the problem effectively, the team was able to understand which components were absolutely necessary for the design to be possible and which were not, such as the motors and the dolly. Then, the team could begin to assess those pieces to figure out what components would work well with them. As the winter semester comes to a close, the team will continue to buy and assess the second round of parts. At the very end of the winter semester, the assembly design for the robot should be complete, and the assembly can begin over winter break and into the beginning of next semester. The hardware and software system will be assembled individually, and the months of February and March will be spent cohesively combining the two systems into the minimum product for the project. The final month of our project, April, will be spent working on the team's stretch goal of fully autonomous navigation. The team is confident the timeline can be followed effectively and that the robot will meet the client's expectations.

6.1 - Gantt Chart

This Gantt chart (see Fig. 3) represents the flow of this project. The three phases, planning, testing, and assembly, are broken into different colors and sections. The vertical red line represents where the team is currently in the timeline, and phase completions are displayed by both the diamonds and text boxes above. These phase completion indicators represent our team's milestones. Though this chart is obviously subject to change, especially on exact dates, the overall progression should remain constant.

Name	Begin date	End date
Team Startup	9/18/18	10/23/18
Tech Feasibility Document	10/24/18	11/9/18
Requirements Document	10/24/18	12/10/18
Planning Phase Complete	12/11/18	12/11/18
Parts Order P1	10/5/18	10/23/18
Hardware/Software Testing P1	10/24/18	11/15/18
Parts Order P2	11/16/18	12/1/18
Hardware/Software Testing P2	12/2/18	12/16/18
Assembly Design	12/17/18	12/31/18
Testing Phase Complete	1/1/19	1/1/19
Assembly	1/1/19	2/12/19
Hardware & Software Integration	2/13/19	3/18/19
Stretch Goal Navigation	3/19/19	5/6/19
Assembly Phase Complete	5/7/19	5/7/19

Figure 3. Gantt Chart for Keystone Robotics.

7.0 - Conclusion

Tours at the Engineering are time consuming and uninformative. Professors must take time out of their day to assist with these tours, and the students who give them are often not engineering students. Our client, a computer science professor with a physics and engineering background, wants a device that can take over this role. By automating this process with a robot-assisted tour, faculty can save time, and visitors to the building can be shown the capability of NAU's engineering and computer science students.

The scope of this project is limited to building a solid and expandable base for other students to work on for future capstones. This base design of a robot must be able to navigate a floor of the engineering building successfully at walking speed. Other requirements our client had were that the specifications and design process for this robot should be well documented and the final product must be usable by a computer science professor. This means the design must be expandable for future use by other capstone teams and students to contribute to the project's end goal and to be used by our client at the end of this capstone project.

The solution that this team has come up with for this capstone is a "30-gallon robot" that uses two microcontrollers for data processing and control of the system while using motors, wheels, and driver circuitry to physically move the device, all housed in or on a 30-gallon barrel. This device should be driven by a large battery or batteries that our client should be able to easily charge every evening. Different arrays of sensors can send data to the microcontrollers for feedback to the robot on its movement and how to adjust the microcontroller output, creating an effective control loop. For the minimum product of this project, this robot should be able to be manually controlled by a controller or joystick module.

This document outlines this team's vision for a solution, requirements that have been discussed with the client, both functional and non-functional, as well as the risks associated with implementing a robot of this scale with unique safety and navigation requirements. With each of the risks discussed, several strategies designed to mitigate each of them have been listed as well.

So far, the team has discussed the client's ideas and requirements with him multiple times, and the technical feasibility and parts acquisition and testing have gone very well. The team has been able to acquire multiple parts as well, including the barrel itself, a dolly, IR sensors, batteries, motors, wheels, and motor mounts. This has allowed various parts testing to begin. The team has also successfully designed a prototype robot, which is much smaller than our planned device and uses ROS to move using a controller, and have done successful research on various ROS libraries. With proper heat mitigation, effective use of sensor data, and well-designed code and circuitry, this team feels confident that the final product will meet the client's minimum requirements. We believe our robot will be able to successfully navigate a floor of the engineering building, and make good progress towards this project's stretch goal of fully autonomous navigation.

Appendix

Risk:	**Overheating**	**Combustion**	**Loosening/ Detaching Components**	**Battery Leakage**
Likelihood	3	1	1	1
Concern Level	Mid	High	Low	Low
Mitigation Strategy	1) Install heat-resistant shelving in interior 2) Cut holes to facilitate airflow for cooling 3) Heat sinks	1) Use rechargeable batteries 2) Heat Sinks & Good Ventilation	1) Order custom mount for motors 2) Perform exhaustive stress tests on assembled product	1) Proper heat dissipation 2) Purchase Lithium-ion 3) Keep records of last charge, replacements

Table 1: Hardware Risks Associated with Robot-Assisted Tours

Risk:	**Localization Errors during Navigation**	**Speed Calculation error**	**Loss of control (manual or automatic)**	**Collision with Passerby or Obstacles**	**Falls/ Failure to Avoid Drop-offs**
Likelihood	3	2	2	4	2
Concern Level	Low	Low	High	Mid	High
Mitigation Strategy	1) User ability to correct robot location in map 2) Ability to halt self & emit light or sound warning	1) Controlled - environment testing of robot movement at variable speeds before deployment in public space	1) Physical emergency stop button (cuts power supply) - Large and easily reachable	1) Sensors placed to minimize blind spots 2) Proximity warning lights & sounds 3) Physically stable system	1) Test to ensure sensors notice drop-offs in manageable amount of time 2) Program stop points/ warnings into map data

Table 2: Software Risks Associated with Robot-Assisted Tours