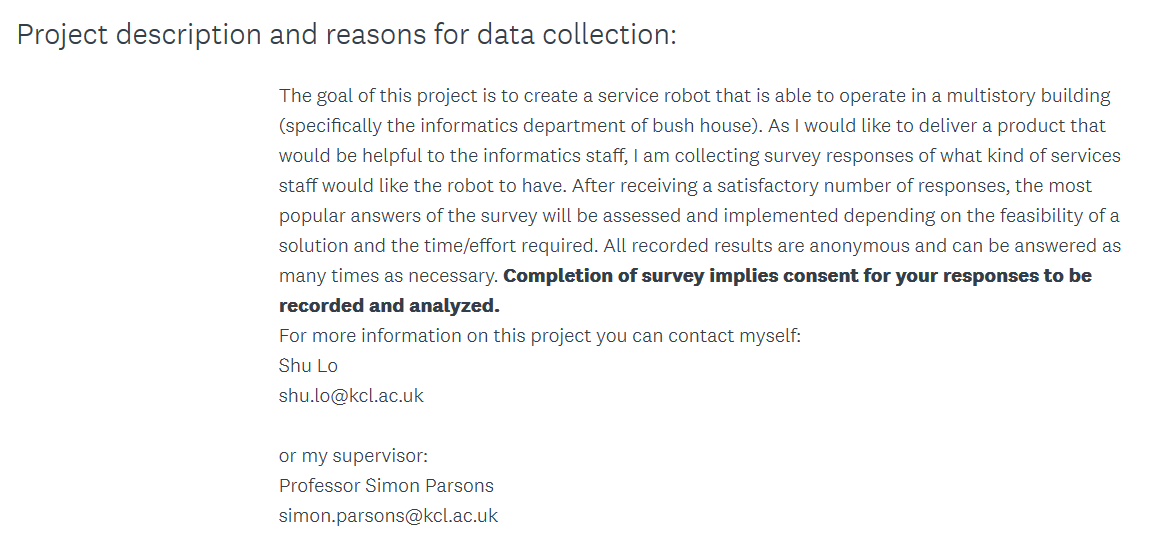
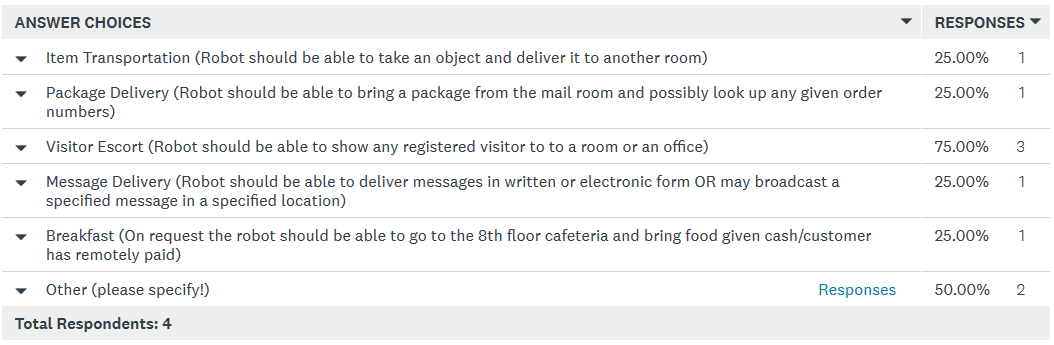
**Background and Context for the project**

The focus of this project is to create a mobile service robot that is able to operate on a multistoried building. This project is important as it explores potential relationships between humans and robots and how each participant can help the other in an office setting. A human-robot team can help increase the efficiency of each participant by correctly allocating the tasks to suit each others’ needs. The starting point of this project is to determine the needs an office has that a robot could fulfill.

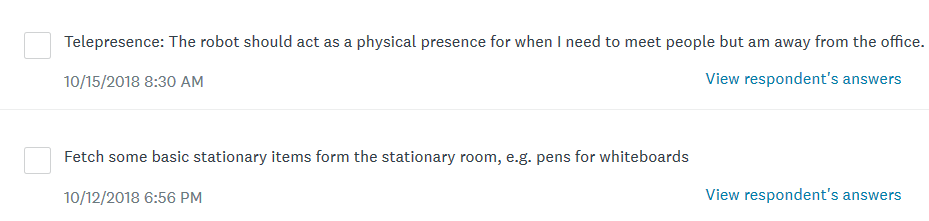
Since this service robot is meant to operate in the Informatics department of Bush House, I decided to survey the Informatics staff on what their preferences for the robot’s functionality should be. A minimal ethical clearance form was approved for this research and a notice of participation implying consent was displayed before the survey questions:



All survey responses were anonymous and staff members were allowed to select more than one option, or fill in an “Other (please specify!)” option. The results are as follows:



The above five choices were created to reflect a variety of potential office needs. Any “Other (please specify!)” options are as follows:



The original goal was to create a decision matrix comparing the popularity and difficulty of implementation of each possible functionality in the answer choices. Any functionality that required a fine amount of physical manipulation with little force (for example, the Breakfast option may have required a robotic arm to grab items or some additional way to maintain the balance of any carried objects while it moves) would be considered a task with a high difficulty of implementation. However, the Visitor Escort option, which required no additional physical functionality, received the majority of the votes. The requirements of this project have now been focused on implementing this one functionality.

**Literature Review**  
  
 I mainly focused on available papers published concerning mobile service robots and a robot project I did last year for a class.

My first paper of reference was *CoBots: Collaborative Robots Servicing Multi-Floor  
Buildings[*2] .This paper details the progress in designing multi-story service robots that are able to perform a series of tasks and collaborate together. The CoBots detailed in this paper are able to escort a visitor to a specified location. Interestingly, the paper also details the CoBots ability to perform a semi-autonomous Telepresence task that allows a human to control the robot. Since one of the “Other (Please Specify!)” options requested a telepresence feature, I may implement this as an additional feature given I have time left over from completing the Visitor Escort functionality. Interestingly, the CoBot’s Visitor Escort ability was the first functionality listed in the paper completed. This, along with the survey responses, may suggest that Visitor Escort is not too challenging to implement or would be a desirable feature in a mobile service robot. Due to the short nature of the paper and the fact that it mainly summarized progress in research and implementation rather than going into detail about actual methodology of how the robots decide to complete their tasks, this paper was mainly useful for the references and ideas in its citations.

Since my project has some focus on the relationship between a mobile service robot and people, most of my literature review mainly focused on how to develop an effective, symbiotic human –robot relationship. Explored aspects of this potential relationship included how robot-human interaction can help increase the efficiency of completing a task. As detailed by S. Rosenthal[1] , human help can allow a robot to increase its capability by performing additional tasks for the robot. A human can help by limiting specifically the physical tasks the robot has to perform and reducing the uncertainty of its environment and state. A robot can plan to call for human help if it knows it is going to encounter one of these situations during task completion. This paper[1] details a CoBot that is able to navigate to a supervising human’s office to ask for help and is aware of the fact that a supervising human it is trying to seek may not be around. In the case of the robot for this project, the robot should just send an email to either me or someone else capable of supervising the robot. Though Rosenthal[1] also mentions that the CoBot is able to ask bystanders for help, I do not wish to implement this functionality in my project as someone may decide to give the robot incorrect information. Interestingly, the paper[1] states that the CoBot has difficulty perceiving chairs and has a resulting increase in navigation time when encountering unexpected objects and that its lack of arms makes most physical tasks such as pushing elevator buttons or picking up mail impossible without human help. As a result, I am planning to establish some kind of communication between the robot and any potential visitor to clarify any needs a robot may have.

Rosenthal[4] has another in depth article detailing an effective human-robot interaction. This paper also details a wifi-based localization algorithm. The robot will occasionally require human help as it may become confused about its localization. The CoBot detailed in this article uses planning to determine its actions and each action has a probability function of success or failure in completing the action. While I initially thought that this article would be helpful since it detailed the states and actions defined in the CoBot’s programming, I have decided not to use planning (see **Changes in Plan** section). This paper also gives a brief description of the wifi-based robot localization and navigation algorithm and shows how different amounts of human help for localization affect the robot’s navigation. The location of the robot is determined using a particle filter and the inferred location of the robot is determined by clustering the particles and taking a weighted mean of the best cluster. I am still not entirely sure how, if possible, to implement my robot’s localization, though it may be possible for my robot to also used some kind of wifi-based localization within Bush House.

Two other papers, one detailing how a robot can proactively seek help from humans[5] , and one detailing gradient refinement for mobile robot localization[3] were also reviewed, but did thus far have not helped much in this project.

Finally, a project [6] I worked on last year may shed some insight on the software aspect of this project. As described in the project ReadMe on GitHub, the project I worked on last year was a robot capable of navigating to a goal location given its starting point and a robot readable version of a map of the course. The programming was done on an EV3 Lego Robot and implemented in Java. I wrote the A\* algorithm used to compute the path the robot should take to reach the goal position. The robot then used differential drive to calculate the speed and direction it needed at various points of the path to reach its goal location. The robot had two sensors –a gyrosensor to determine its relative direction from its starting point and a color sensor. The color sensor was used to read some colored pieces of tape in the arena to help the robot’s localization. Unfortunately, I did not work on the localization and the robot was also unable to localize correctly during the test run. However, creating a visual print out of the path on the map on the console showed that the A\* algorithm worked perfectly for finding a path to the goal. I have decided to use an A\* algorithm in my implementation of this project.

**Requirements**

This project has several base requirements that must be fulfilled regardless of the functionality decided by the outcome of the survey. The robot is expected to:

1. Identify its location on a floor (Localization)
2. Be able to navigate the informatics department floors (5 to 7 of Bush House)
3. Identify if there is an object in its path
4. Be able to contact a human that can help in an unexpected situation

Any present human companion will also need to be able to provide appropriate help and requests. These are as follows:

1. Open any doors to allow the robot to pass
2. Push any buttons to call an elevator or stop on a floor number
3. Request an escort from the robot to a location in the Bush House Informatics Department
4. Possibly move any small objects out of the way of the path of the robot as not to increase navigation time

There exist some requirements specific to the Visitor Escort functionality for the robot:

1. The robot must be able to understand the needs of a visitor
2. The robot must be able to navigate to a visitor specified location

Some possible requirements may be needed to insure the robustness of the functionality. I define robustness as additional implementation needed to help guide the robot in case of an unexpected situation. If there is enough time to implement these requirements, the robot should be able to:

1. Determine which floor the elevator has opened on and whether this is the floor the robot has been instructed to go to
2. Navigate around an unexpected, seemingly unmoving object in its path
3. Return that a room requested by a visitor is not available
4. Use a lack of reply to deduce that the visitor has left

All programming for this project is to be done in ROS, a middleware language for robotics. Some ROS Packages I have decided to use are detailed in the **Specification** section below. The first versions of the code will be written in ROS Lunar and may need to be translated to ROS Melodic to run. I also require the floorplans of the Informatics Department to create a robot readable map in ROS. Most of the detail in the map will be irrelevant since the robot should not be entering any rooms. The most important details of the floorplans the robot requires are the locations of the offices, toilets, elevators, and any unmoving obstacles that could be a cause for concern if not correctly added. As of today, the only obstacle I have had to add to the floorplans is the directions block in the middle of each lobby.

Some test cases will also be required once the robot implementation has been completed to correctly tune the robot’s movements. The robot will be given a series of locations it will need to take a visitor. Depending on the amount of implementation completed before the test cases, the visitor may also be instructed to walk away in the middle of an escort or intentionally choose the wrong button in an elevator to see how the robot reacts.

Due to the fact that live robot demonstrations do not always go as planned, these test cases will also need to be filmed as recorded evidence that the robot is able to perform. I will also need to write several reports on my progress in this project, this paper being one such report.

**Specification**

**Hardware**

The robot used for this project is the Robotino – For research and education. Festo Didactic’s description of the Robotino states that it is able to move in four directions and able to rotate in place. The Robotino also has a CPU, though it may be more feasible to attach a laptop to the Robotino as control since a laptop will be able to display information and therefore communicate with a human.

Some additional hardware will be needed to support the robots functionality. Due to the fact that the robot must be able to determine if an object is in front of it, the robot may need a sensor to determine the distance to a close object, such as an ultrasonic sensor. Besides allowing the robot to determine if its path is blocked, the sensor should also help the robot determine if an elevator or office door is open.

As I am not entirely sure how the robot will react to small bumps and cracks, an additional sensor may be needed to assess whether a gap (for example, the gap between the elevator and the floor when the doors are open) is too large for the robot to cross. However, enough tuning and testing may be enough to determine if any such conditions exist within Bush House.

The robot will also need to be able to assess whether it is accelerating upwards or downwards when it expects to be in an elevator, which can be done using an accelerometer. This, along with the amount of time during its acceleration, will allow the robot to determine which floor the elevator is at when it has stopped. As I do not believe it feasible for the robot to try and calculate how high the elevator has ascended or descended given that accounting for real world physical forces (for example, air resistance) are difficult, I have decided to instead take the time to write code to generate the amount of time the robot finds itself accelerating and determining a time range of travel between floors. With a sufficient amount of data, the robot should be able to estimate how much time it takes to get to a floor, and should therefore be able to tell if the elevator opens on the right floor.

An accelerometer may also have some additional functionality to the robot in determining accelerations in cardinal directions to provide the robot some idea on where it should be in the building. I am not entirely sure if I am going to implement this functionality, however.

The robot must also have a wireless network adapter and access to the wifi within the informatics building to be able to send emails. If the telepresence functionality is implemented, this will also allow an informatics staff to remotely control the robot. The robot will need to use the laptop’s camera to present to the staff a view of the building from the robot.

**Software**

As stated in the requirements, all programming for this project will be done in ROS. I have decided on several ROS Packages to implement my solution. Due to the limited capabilities of the laptop I am mainly programming on, I am using ROS Lunar instead of ROS Melodic (the latest version of ROS).

Since I am planning on using an accelerometer to determine the upwards and downwards acceleration, I am planning on using a dmu11driver and its respective package. I have chosen this particular accelerometer because it is compatible on ROS Lunar and Melodic and is also a lightweight, relatively low- cost accelerometer.

ROS has a navigation package that I will use to create a robot readable map.

I am still undecided on how to determine which elevator the robot should take.

Below is a goal outline of the robot’s course of action when it receives an escort request from a visitor. Some of the functionality is not yet applicable but could be implemented given enough time:

1. Receive room instructions from the visitor.
2. Check if the room is available/on the map.
   1. If the room is not available, state so to the user.
3. If the room is on the same floor as the robot:
   1. A\* to appropriate lobby door.
   2. Ask visitor to open door.
      1. If the visitor does not open the door within 15 seconds, assume the visitor has left.
   3. A\* to requested room door.
4. If the room is not on same floor:
   1. A\* to the elevator.
   2. Ask for visitor to press appropriate arrow button.
   3. Enter elevator when door opens.
   4. Use accelerometer to make sure the elevator stops on the right floor.
      1. Assume the visitor stays in the elevator until the door opens on the right floor.
   5. Wait for the elevator door to open.
   6. Exit the elevator.
   7. A\* to appropriate lobby door
   8. Ask the visitor to open the door.
      1. If the visitor does not open the door within 15 seconds, assume the visitor has left.
   9. A\* to room door.
5. Ask for if the visitor is at the right place.
6. If yes or no reply within 15 seconds go back to the lobby door.
   1. If the reply is no, ask the visitor to help localize.

**Changes in Plan**

As I am taking a course on Artificial Intelligence Planning and am interested in applying the class to this project, I originally planned to use ROSPlan in my implementation. However, watching a failed live demonstration and the fact that I am unable to enable virtualization on my laptop has convinced me to use standard ROS packages instead.

I originally intended to use the ROS gmapping package to translate the floorplans to a robot readable map. However, gmapping is not supported past ROS Lunar, so this would be a problem if I decide to run the robot with ROS Melodic.

For my current plan of implementation, I have decided to limit the locations a visitor may request an escort to the either a room number, name of a professor or office room number, computer lab, or a specified toilet (male, female, disabled).

I also, as of now, currently behind schedule in my implementation of this project due to the fact I did not account for the time I would need to complete job applications. As a result, code for the robot path planning has not yet been completed.

**Cited Papers and Works**

**1**S. Rosenthal, M. Veloso, and A. Dey, ―Task behavior and interaction planning for a mobile  
service robot that occasionally requires help, in Workshops at the Twenty-Fifth AAAI  
Conference on Artificial Intelligence, 2011.

**2**M. Veloso, J. Biswas, and B. Cotlin ‖CoBots: Collaborative Robots Servicing Multi-Floor  
Building IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012  
——, ―Depth camera based indoor mobile robot localization and navigation, in Proceedings of  
ICRA’12, the IEEE International Conference on Robotics and Automation, 2012. <http://www.cs.cmu.edu/~srosenth/papers/12irosvideo-VelosoEtAl.pdf>

**3**J. Biswas, B. Coltin, and M. Veloso, ―Corrective gradient refinement for mobile robot  
localization, in Intelligent Robots and Systems (IROS), 2011 IEEE International Conference on.  
IEEE, 2011 (PDF) CoBots: Collaborative robots servicing multi-floor buildings.

**4**S. Rosenthal, J. Biswas, and M. Veloso, ―An effective personal mobile robot agent through  
symbiotic human-robot interaction,‖ in Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1. International Foundation for  
Autonomous Agents and Multiagent Systems, 2010, pp. 915–922. <http://www.cs.cmu.edu/~mmv/papers/10aamas-cobot.pdf>

**5**S. Rosenthal, M. Veloso, and A. Dey, ―Is someone in this office available to help me?‖ Journal  
of Intelligent & Robotic Systems, pp. 1–17, 2011. <http://www.cs.cmu.edu/~mmv/papers/11jirs-stephanie.pdf>

**6**Shuistlo, -- Robot Obstacle Avoidance to Reach a Goal. Code and Video available at <https://github.com/Shuistlo/Robot-Obstacle-Avoidance-to-reach-a-Goal>