Bilkent University
Department of Computer Engineering

Senior Design Project

T2325

LibreBot

Detailed Design Report

21902298, Giray Akyol, giray.akyol@ug.bilkent.edu.tr
21901779, Muhammed Can Küçükaslan, can.kucukaslan@ug.bilkent.edu.tr
21802880, Muhammet Hikmet Şimşir, hikmet.simsir@ug.bilkent.edu.tr
21803473, Mustafa Utku Aydoğdu, utku.aydogdu@ug.bilkent.edu.tr
21802530, Mustafa Yasir Altunhan, yasir.altunhan@ug.bilkent.edu.tr

Supervisor: Salih Özgür Öğüz
Course Instructors: Erhan Dolak, Tağmaç Topal

2023-03-11

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Senior Design Project course CS491/2.

Contents

Detailed Design Report

LibreBot

1 Introduction

Organizing books in a library and keeping them organized is a tedious and time-consuming job that humans still perform. The time and effort put into finding a desired book in the library by a library user increases as the number of books and categories increase in the library. In the age of the fourth industrial revolution, where autonomous robots became a significant part of the workforce in many sectors, the use of robots in this daunting job is still very limited. Our project's purpose is to model a robot that can be used to automate the organization of library books taking advantage of state-of-the-art technologies. The robot will be able to perceive its environment and detect books using computer vision, identify books by radio frequency identification (RFID) tags or call numbers, and collect from or place them on appropriate shelves, according to the library classification system, using the state-of-the-art machine learning technologies for task and motion planning (TAMP).

1.1 Purpose of the system

Our project's purpose is to model a robot that automates the process of placing a delivered book on the shelves, finding and bringing the desired book to the user, and maintaining the book order of the library. When a book is returned the robot will be able to place the book in the correct shelf in the library. The robot also can collect unattended library books from the study desks and put them on the appropriate shelves. Besides those, it will be able to bring the desired book that was specified by its "call number" from a user. To specify a call number to the robot, the user will log in his/her library account, assuming the library management integrated control instructions of the robot to their library website with our help, and add its desired book to the queue of books to fetch. The robot will also scan through the shelves to check whether there are any books placed on wrong shelves or in wrong order regularly -with time interval specified by the library personnel. The library management and a normal user will have different privileges for the control of the robot. For instance, the library manager may command the robot to look for the shelves and adjust the order of the books if their order is not correct whereas a normal user may only be authorized -indirectly– to issue a book fetch request in the system.

Normally, building such a robot involves heavy work for both hardware and software components. However, our project is focused on the software part of the robot that involves but is not limited to computer vision, object detection, object recognition, decision-making for both low and high-level decisions, motion planning under specific constraints (either using solvers for linear and non-linear optimization problems or using reinforcement learning). We will use the drake robotics framework [1] to design the robot model, to simulate the behavior of the robot, to simulate the 3-D environment involving robots, bookshelves, books, desks, people, and other 3-D objects, and to test the functionalities.

1.2 Design goals

The LibreBot should be compatible and complementary with the existing library system:

- To avoid duplication or inconsistency of data between the robot and the library system
- To enable seamless communication and coordination between the robot and the library staff
- To enhance the efficiency and accuracy of the library operations
- To improve the user experience and satisfaction

The LibreBot should be reliable, safe and secure:

- To prevent malfunction or breakdown of the robot that could disrupt or damage the library services
- To protect the robot from unauthorized access or misuse that could compromise its functionality or data

- To ensure the safety of users and staff who may encounter or interact with the robot
- To comply with ethical and legal standards for using robots in public spaces

2 Current System

AuRoSS is a system proposed by Li, Huang, et al. [6] that strives to solve the library book tracking problem LibreBot also aims to solve. A significant difference between LibreBot and AuRoSS is how the systems identify books. While both LibreBot and AuRoSS use RFID tags, LibreBot will also use computer vision and text recognition when RFID is not available. If the book does not have RFID tags LibreBot requires that the book label is not damaged, not obstructed and the environment has sufficient lighting to extract the information. On the other hand, RFID tags require high-accuracy positioning of the scanner and enough range of the scanner [6]. In general, libraries already have labels on books but RFID tags are new technology (by library standards). While scanning RFID tags may be quicker, to use AuRoSS, libraries must tag the books, which costs time and money. Compared to AuRoSS, LibreBot is an end-to-end system, it handles the book-handling process from an electronic book request to its delivery. AuRoSS only handles missing and misplaced books while LibreBot handles gathering the books from shelves, placing back borrowed books, and correctly sorting misplaced books. LibreBot's scope is much larger and it is a more complete system compared to AuRoSS.

3 Proposed System

3.1 Overview

Our project's purpose is to model a robot that automates the process of placing a delivered book on the shelves, finding and bringing the desired book to the user, and maintaining the book order of the library. When a book is returned the robot will be able to place the book in the correct shelf in the library. The robot also can collect unattended library books from the study desks and put them on the appropriate shelves. Besides those, it will be able to bring the desired book that was specified by its "call number" from a user. To specify a call number to the robot, the user will log in his/her library account, assuming the library management integrated control instructions of the robot to their library website with our help, and add its desired book to the queue of books to fetch. The robot will also scan through the shelves to check whether there are any books placed on wrong shelves or in wrong order regularly -with time interval specified by the library personnel. The library management and a normal user will have different privileges for the control of the robot. For instance, the library manager may command the robot to look for the shelves and adjust the order of the books if their order is not correct whereas a normal user may only be authorized -indirectly– to issue a book fetch request in the system.

Normally, building such a robot involves heavy work for both hardware and software components. However, our project is focused on the software part of the robot that involves but is not limited to computer vision, object detection, object recognition, decision-making for both low and high-level decisions, motion planning under specific constraints (either using solvers for linear and non-linear optimization problems or using reinforcement learning). We will use the drake robotics framework [1] to design the robot model, to simulate the behavior of the robot, to simulate the 3-D environment involving robots, bookshelves, books, desks, people, and other 3-D objects, and to test the functionalities.

One of the innovation types that our project is related to is service. After our project is finished and implemented in a library system, a user of a library will no longer need to look for the exact location of a book in the library system, walk into the chamber shelf where the desired book is located, take the book and register the book into the library system but simply issue a fetch request of a specific book in the library system. Therefore, it will ease the job of a user.

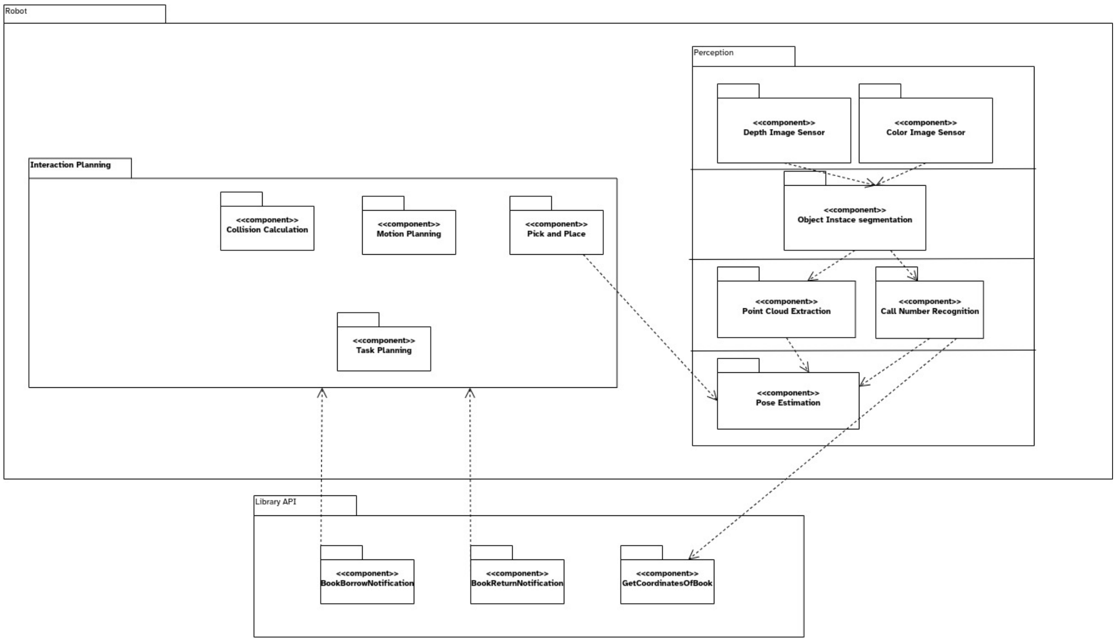
From time to time, users of the library may put a book on the wrong shelf or position so the library personnel may need to walk around and check the shelves to check if there are any books incorrectly located and put it into the correct position.

However, with the help of the robot, this task of the library personnel may be delegated to the robot which makes the life of the library personnel much easier.

The scope of the change of our project is transformation rather than optimization because its main aim is not to improve the productivity of an already existing mechanism of fetching and placing library books but to propose a totally new solution that take advantage of the state-of-art autonomous robot technologies to place returned books into the shelves, to fetch desired books from the shelves and to keep the correct order of the books in the shelves.

To illustrate how the robot works, consider a case where the robot places a book left on a table into the correct position of the library. First, the robot moves around the library to find any book left on a desk. When the robot encounters a desk with no one using it for at least 45 minutes (the amount of time can change according to a specific library rule), it stops next to the desk and runs an object detection algorithm for the items on the desk. The reason why the robot runs an object detection algorithm is that it should not pick a random object such as a pencil or water cup, but pick a book. After the execution of the object detection algorithm, it should pick any one of the books left using a motion planning algorithm to be able to grasp the object successfully without harming the environment. For instance, there may be people nearby so the robot should have a restricted working area and restricted angles for the movements of its arms in order not to hit any person or object nearby. Besides that, the book may not easily be grasped since there is no space between the book and the desk so the robot may move the book to a corner of the desk and grasp from there which requires different types of 6 or fewer degrees of freedom [2] in different parts of the motion. After that, an object recognition algorithm is run to check whether it is a user book or a library book. The plan, for now, is that the internal structure of the object recognition algorithm will benefit from the area of computer vision and look for a “call number” written on the book which is used to identify library books. After that, the robot needs to go to the correct position of the book in the library, which may or may not require the robot to go to a different floor or chamber. For the case of the robot going to a different flat, decision-making is used to decide which path to take in order to go to the specific flat and chamber. Feedback control is necessary to make sure the robot goes to the right place without harming or restricting its environment and itself. Each step or movement may give feedback to the robot so that it takes a better-optimized step or motion in consecutive movements. After finding and going to the right flat and chamber, the robot places the book into the correct position again using decision-making, motion planning, and object recognition (to find the exact spot of the correct position of the book by looking at the nearby books).

3.2 Subsystem decomposition



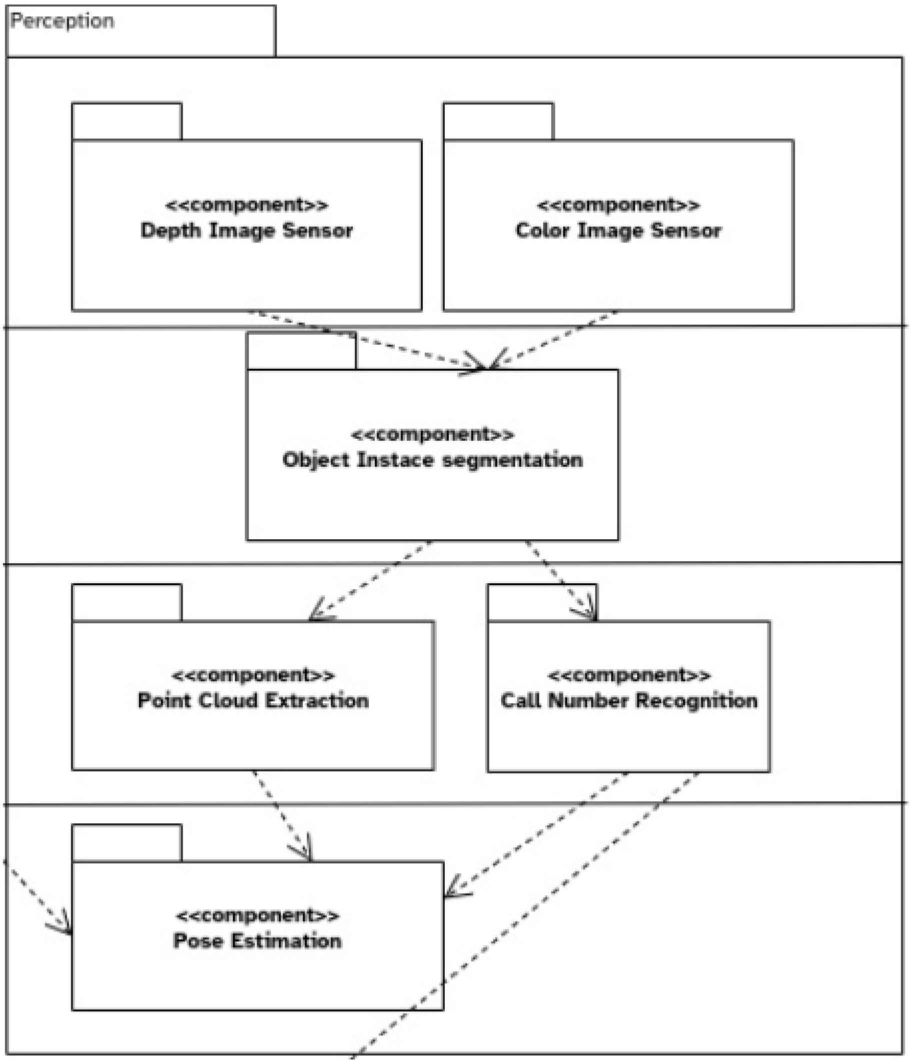
Based on functional requirements we decided to decompose our robotic software into two subsystems: Interaction Planning and Perception.

The Perception subsystem is designed to get data from the world and process it into a structured representation for other subsystems. It uses a layered architecture to organize data flow between components. Data comes from two sensors: Depth image sensor and Color image sensor. Sensor data flows into the segmentation subsystem which transforms it into a set of cropped images of objects such as book, shelf, obstacle, etc. In the next layer the images are fed into point cloud extraction which converts the 2D representation into a 3D representation. The call number recognition system extracts the call number of the book. The point cloud and call number are fed into the Pose estimation subsystem to calculate the pose of the objects in format usable by the interaction subsystem, this subsystem may also make a request for the coordinate information about this book to the library API.

Interaction Planning subsystem is designed to determine how the robot should perform in the physical environment. It has four components. Collusion calculation component is used to extract the collusion space of the environment given the point cloud of the environment as input. Motion planning is used to determine the optimal path that the robot should follow while avoiding the collusion space. After the robot follows an optimal path and arrives at a shelf, the pick and place component is used to determine the pre-pick and pre-put places of the book and to specify the necessary interpolation functions between the points. Task planning component is used to determine the order of commands to achieve an objective such as going from an initial point to the shelf where the desired book is located, picking the book and taking the book back to the initial point.

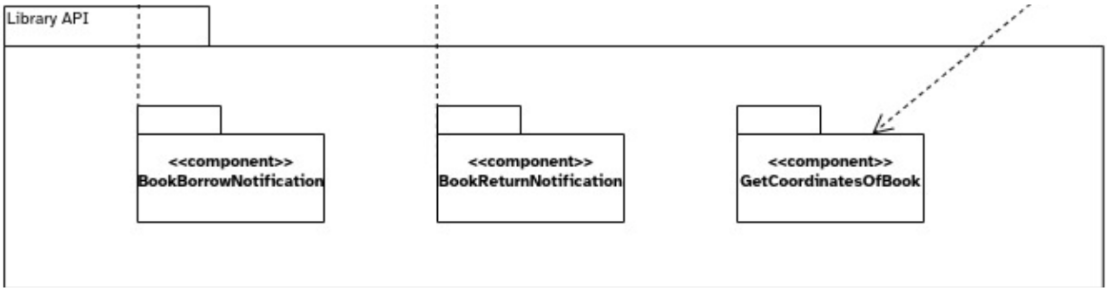
4 Subsystem services

Perception



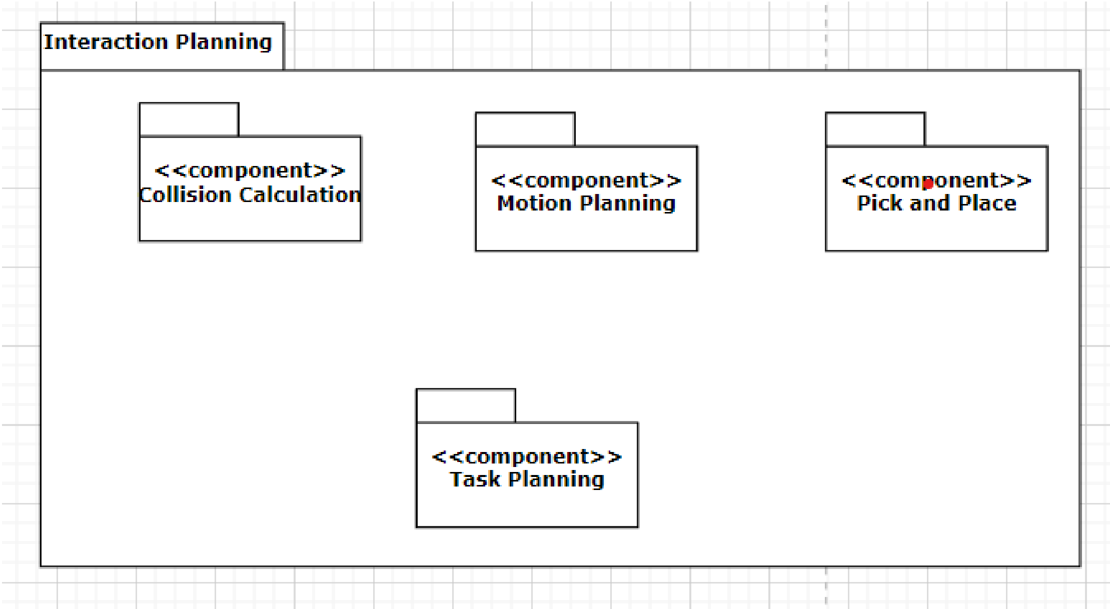
- Depth Image Sensor: Gather distance data and outputs a 2d array of depth metrics.
- Color Image Sensor: Gather color data and outputs a 2d array of RGB values.
- Object Instance Segmentation: Using ML extracts instance masks from the color image data and uses it to focus depth image to a certain object.

- Point Cloud Extraction: Converts masked 2D depth image to color points in 3D space.
- Call Number Recognition: uses OCR algorithms on the masked color image to recognize book call numbers.
- Pose Estimation: Using point cloud data iteratively calculates the 6 degree pose of the object for use by interaction subsystems.



- BookBorrowNotification: When a user requests a book to be delivered this subsystem notifies the robot about the request.
- BookReturnNotification: When a user returns a book to the library this subsystem notifies the robot about the request.
- GetCoordinatesOfBook: Given a call number responds with where the book should be placed.

Interaction Planning



- Collusion Calculation: Using point cloud data obtained from the environment, it creates the collusion space.
- Motion Planning: Avoiding the collusion space and given the objective point for the robot to arrive, it calculates an optimal path.
- Pick and Place: Given initial and target pose of a book or object, it calculates the pre-pick, pre-put places and makes necessary interpolation between the desired points.
- Task Planning: It is used to specify the order of commands to give to the robot when the robot gets book borrow or book return notification.

5 Test Cases (functional and non functional test cases, procedures and expected results)

5.1 Image Segmentation Test Cases

In image segmentation, our main purpose in this project is to partition the images we take from the cameras inside the simulation into multiple segments or regions, each of which corresponds to a different object or region in the environment. The goal of image segmentation is to simplify and/or change the representation of an image into something more meaningful and easier to analyze. In our project, the main purpose is to extract depth images of the object around the robot, and take the corresponding action. One particular demonstration of this can be seen in the below figures. Our objective is to make a transition from the RGB Image on the left to the segmented image on the right.

During the image segmentation process, some of the faced problems are as following :

1) Ambiguity and uncertainty: Images can be ambiguous or uncertain, and the algorithm may struggle to segment them accurately. For example, objects with similar color or texture can be challenging to distinguish, and images with noise or low contrast can be difficult to segment.
2) Lack of labeled data: Image segmentation algorithms typically require labeled training data to learn the features and patterns of objects in the images. However,

obtaining labeled data can be expensive and time-consuming, and may not be available for all types of images or applications.

3) Complex object shapes: Objects with complex shapes, such as those with concavities or non-convex boundaries, can be challenging to segment accurately. The algorithm may also struggle to distinguish between objects that overlap or touch.

In our project, some of the problems are as following:

- Test case 1: Segment image of bookshelf
    - Test objective: Verify that the robot can segment an image of a bookshelf into different regions or pixels that correspond to different books or objects, and get the extracted depth images.
    - Test input: An RGB image of a bookshelf with various books and objects on it.
    - Test output: A segmented depth image with each region or pixel labeled with a unique identifier.
    - Test criteria: The number and shape of regions or pixels should match the number and shape of books and objects in the input image.

- Test case 2: Assign label to each region or pixel
    - Test objective: Verify that the robot can assign a label or class to each region or pixel based on its appearance, shape, color, texture, etc.
    - Test input: A segmented image of a bookshelf with each region or pixel labeled with a unique identifier.
    - Test output: A labeled image with each region or pixel assigned a label such as book or non-book.
    - Test criteria: The labels should be correct and consistent for each region or pixel according to its appearance and features.

- Test case 3: Identify and classify all books in image
    - Test objective: Verify that the robot can identify and classify all books in an image regardless of their orientation, size, position, etc.
    - Test input: A labeled image of a bookshelf with each region or pixel assigned a label such as book or non-book.
    - Test output: A list of books identified and classified by their title and author from the input image.
    - Test criteria: The list should contain all the books in the input image and their correct title and author.

- Test case 4: Identify all non-book objects in image
    - Test objective: Verify that the robot can identify all non-book objects in an image such as book shelves that books are placed in, etc.
    - Test input: A labeled image of a bookshelf with each region or pixel assigned a label such as book or non-book.
    - Test output: A list of non-book objects identified and classified by their type and category from the input image.
    - Test criteria: The list should contain all the non-book objects in the input image and their correct type and category.

- Test case 5: Handle different types of images that contain books or other objects in a library
    - Test objective: Verify that the robot can handle different types of images that contain books or other objects in a library.
    - Test input: Different types of images that contain books or other objects in a library.
    - Test output: Segmented, labeled, identified and classified images for each type of input.

- Test criteria: The results should be accurate and consistent for each type of input.

- Test case 6: Deal with noise, camera position and angle changes that may affect the quality and consistency of image segmentation and recognition
  - Test objective: Verify that robot can deal with Deal with noise, camera position and angle changes etc. that may affect the quality and consistency of image segmentation and recognition
  - Test input: Images with different levels of noise, Deal with noise, camera position and angle changes , etc.
  - Test output: Segmented, labeled, identified and classified images for each level of difficulty.
  - Test criteria: The results should be robust and reliable for each level of difficulty.

5.2 IRIS & GCS Test Cases

Trajectory calculation can be performed with the utilization of the IRIS and GCS algorithms together. Given an obstacle in an area, calculating the path directly from one point to another without collision can be hard to calculate. There is a risk that there is an optimal solution since the area is not convex. To overcome this, the IRIS algorithm partitions the area into convex regions. In such convex regions, optimization problems are much easier to deal with. To make such partitions some seed points should be given to the algorithm. Testing the training of the algorithm with different seed points gives a better understanding of the IRIS algorithm and optimal seed points can be found. This makes the trajectory optimization problem easier to solve.

Test case 1: Testing the effect of different seed points on the IRIS algorithm.

- Test objective: Finding the optimal seed points for a particular simulation setting. Plus, minimizing the training time of the algorithm.
- Test input: Seed points and simulation environment
- Test output: Set of partitioned regions

- Test criteria: The set of convex regions are generated with respect to the seed points within a reasonable time interval.

After finding convex regions, a graph algorithm (GCS) is used to find the least cost solution. This algorithm requires the joint positions of the robot at the start and end of the trajectory. Testing for different joint positions reveals that some joint positions lead to not connected graphs and there is no solution for such cases. Thus, for this algorithm, the joint position you provide to the algorithm determines the success of the test cases.

Test case 2: Testing the effect of different joint positions provided to the GCS algorithm.

- Test objective: Calculating a smooth trajectory
- Test input: Simulation environment, regions created from the previous procedure, and joint position of the robot for the start and end of the trajectory
- Test output: Trajectory and run time of the movement
- Test criteria: A connection is constructed and the minimum cost is calculated.

5.3 User Interface (Website) Test Cases[1]

- Test case 1: The user can search for a book by title, author, genre, or keyword
    - Test objective: To verify that the user can find the book they want using different criteria
    - Test input: A valid title, author, genre, or keyword for a book
    - Test output: The website displays a list of matching books with their details and availability
    - Test criteria: The website returns relevant and accurate results based on the input
- Test case 2: The user can request a book from the robot
    - Test objective: To verify that the user can order a book and get it delivered by the robot
    - Test input: A valid book ID or call number from the search results

[1] We assume that there is already a library website. So we only considered the test cases would be needed for integration of our robot (model) into the existing website/system.

- Test output: The website confirms the request and shows the estimated delivery time and location
- Test criteria: The website sends the request to the robot and updates the status of the book

- Test case 3: The user can cancel a book request from the robot
  - Test objective: To verify that the user can cancel a book order and get a confirmation from the robot
  - Test input: A valid book ID or call number from the request history
  - Test output: The website cancels the request and shows a confirmation message
  - Test criteria: The website notifies the robot and updates the status of the book

5.4 Other Test Cases

- Test case 1: The robot can locate and identify the book by its call number or RFID tag
  - Test objective: To verify that the robot can scan and recognize the book correctly
  - Test input: A book with a valid call number or RFID tag on a shelf
  - Test output: The robot displays or announces the title and author of the book
  - Test criteria: The robot matches the book information with its database
- Test case 2: The robot can grasp and lift the book without damaging it
  - Test objective: To verify that the robot can handle the book safely and securely
  - Test input: A book with a valid call number or RFID tag on a shelf
  - Test output: The robot holds the book in its gripper or arm without dropping or tearing it

    - Test criteria: The robot applies appropriate force and pressure to grasp and lift the book
- Test case 3: The robot can navigate to the destination shelf without colliding with obstacles or people
    - Test objective: To verify that the robot can move smoothly and avoid collisions
    - Test input: A destination shelf number or location for the book
    - Test output: The robot reaches the destination shelf without hitting any objects or persons along its path
    - Test criteria: The robot uses its sensors and map to plan and execute its motion
- Test case 4: The robot can place the book on the destination shelf according to its classification or order
    - Test objective: To verify that the robot can sort and arrange the books correctly
    - Test input: A destination shelf number or location for the book
    - Test output: The robot puts the book on the designated spot on the shelf according to its category or sequence
    - Test criteria: The robot follows library rules and standards for shelving books
- Test case 5: The robot can carry multiple books at once without dropping or damaging them using its storage
    - Test objective: To verify that the robot can transport books efficiently and reliably
    - Test input: A number of books with valid call numbers or RFID tags on a shelf

- Test output: The robot carries the books in its gripper, arm, basket, or tray without dropping or damaging them
- Test criteria: The robot maintains balance and stability while carrying the books

- Test case 6: The robot can check if the given destination shelf or location exists and is accessible
  - Test objective: To verify that the robot can handle invalid or unavailable inputs gracefully
  - Test input: A destination shelf number or location that is either non-existent, occupied, blocked, or out of reach for the book
  - Test output: The robot displays or announces an error message and asks for a new input or returns the book to its original shelf
  - Test criteria: The robot does not attempt to place the book on an invalid or unavailable destination shelf or location
- Test case 7: The robot can distinguish books from other items on the tables such as pencils, pen bags, electronic devices.
  - Test objective: To verify that the robot can recognize and select only books for shelving
  - Test input: A mix of books and non-books items with valid RFID tags on a table
  - Test output: The robot scans and identifies only the books and ignores the non-books items
  - Test criteria: The robot does not grasp or lift any non-books items

6 Consideration of Various Factors in Engineering Design

In this section, many aspects that may affect the LibreBot design will be discussed.

6.1 Public Health Considerations

Since many diseases can be transferred through human contact, it is desired that LibreBot is as autonomous as possible. This decreases the risk of infection. LibreBot should not require the help of library personnel frequently.

6.2. Public Safety Considerations

LibreBot should not make any moves that can damage people or make bookshelves to fall to the ground. So, its vision should be wide enough to see surroundings, and high enough quality to detect objects, especially humans. Pick and place and pathfinding algorithms should be robust and consider any human contact while performing these operations.

6.3. Public Welfare Considerations

There is no direct effect of public welfare that influences the LibreBot design.

6.4. Global Considerations

Since LibreBot will be used by people speaking various languages, it should be designed to work with different languages. Also, vision text extraction should be able to work with different alphabets. Different book ordering systems should be recognized by LibreBot. There should be language support for different languages.

6.5. Cultural Considerations

From culture to culture, the interior design of libraries and bookshelves may change. Path finding algorithms should work with different library designs.

6.6. Social Considerations

Since libraries are social places where people prefer being quiet, LibreBot should not create much noise. To not disturb people, distracting elements such as lights, excessive movement, and vibrations should be minimized

6.7. Environmental Considerations

Since the robot will be active during the working hours of the library and constantly consuming energy, energy consumption should be minimized to make it more environmentally friendly. To achieve this, the design of the robot should achieve the followings:

- Energy-efficient motors and lightweight materials should be preferred. The use of heavy metals that damage the environment should be minimized, recyclable materials should be used instead.
- If the robot has more than one book to place, the robot should cover a minimal distance between bookshelves.
- In order to train the ML algorithms considerable computational power is needed which uses a significant amount of energy in the form of electricity. This energy may come from fossil or non-renewable sources that impact the environment negatively. Inefficient training algorithms and hardware should not be preferred.

6.8. Economic Considerations

Number of library bots in the library should be the minimum number that can satisfy the request and alignment requests in the library. That's why processing and performing requests should be handled efficiently.

	Effect Level	Effect
Public health	2	More autonomy
Public Safety	3	Better camera, pick and place and pathfinding algorithms

Public welfare	0	No effect
Global factors	4	Text recognition will be trained for different alphabets, language support
Cultural factors	1	Pathfinding algorithm trained for various library designs
Social factors	3	Less noise, movement, light, vibration
Environmental Factors	6	Less energy consumption
Economic Factors	7	Efficient request processing and performing the requests

Table 1: Factors that can affect analysis and design.

7 Teamwork Details

It is aimed that every person in the group will contribute to the project. In order to ensure this division of the work with respect to the skills and capabilities of the group members is important.

7.1 Contributing and functioning effectively on the team

- Consulted with the group members and chose the most sensible and popular options together as a group
- Contributed equally to the preparations of reports together as a group
- Participated in the project demo together as a group
- Task and Motion Planning Modules are implemented by Mustafa Yasir with the help of Muhammed and Giray.
- Image segmentation module is designed and implemented by Utku with the help of Giray and Hikmet

- IRIS and GCS module is being designed and implemented by Hikmet with the help of Giray and Utku
- The very first prototype of pick and place demos was implemented by Giray with the help of other members

7.2 Helping creating a collaborative and inclusive environment

Throughout the project, we demonstrated an effort to establish the following:

1) Designating clear and realistic goals and expectations for the project , which can be implemented in the given time and equal amount of workload among the team members. This helped us to avoid any frustration and disappointment we faced during the project phases.
2) We have assigned roles and responsibilities based on each member's interests and strengths. This separation of work based on the member's interests increased the total amount of time each member spent on the project drastically.
3) Although we mostly divided the group work into independent parts in order to increase the parallel efficiency of the group, we did not only focus on our own parts but examined each other's work so that our independent parts are coherent in the overall project and gave feedback to our teammates so that the work of each team member was corrected and improved by other teammates.
4) Throughout the project, we respected each other's opinions and backgrounds, and come into a compromise when different ideas conflicted. We helped each other when a theoretical concept is not understood by one of the team members. Similarly, we helped each other during the implementation phases, which increased the comradeship inside the team.

7.3 Taking lead role and sharing leadership on the team

- The simulation environment was set up by Muhammed Can Kucukaslan and Mustafa Yasir Altunhan.
- Giray Akyol led the research on the algorithms and supported the team in implementation and research.
- Mustafa Yasir Altunhan oversaw the paperwork.

- Mustafa Utku Aydoğdu led the design of ML algorithms for the vision part.
- Muhammet Hikmet Şimşir integrated the vision part of the project and implemented the IRIS & GCS algorithm.
- Muhammed maintained project website

8 References

[1] Russ Tedrake and the Drake Development Team, "Drake: Model-Based Design And Verification For Robotics," 2019. [Online]. Available: https://drake.mit.edu. [Accessed: 16-Oct-2022].

[2] Wikipedia, "Underactuation," Wikipedia, 17-Oct-2022. [Online]. Available: https://en.wikipedia.org/wiki/Underactuation. [Accessed: 16-Oct-2022].

[3] R. Tedrake, "Robotic Manipulation: Perception, Planning, and Control." [Online]. Available: http://manipulation.mit.edu. [Accessed: 16-Oct-2022].

[4] A. Quart, "Automation Is A Real Threat. How Can We Slow Down The March Of The Cyborgs?," The Guardian, 08-Aug-2017. [Online]. Available: https://www.theguardian.com/us-news/2017/aug/08/humans-v-robots-defending-jobs. [Accessed: 16-Oct-2022].

[5] I. Wladawsky-BergerGuest, "Automation And The 2030 Job Hunt," WSJ, 09-Feb-2018. [Online]. Available: https://www.wsj.com/articles/automation-and-the-2030-job-hunt-1518198504?tesla=y. [Accessed: 16-Oct-2022].

[6] R. Li, Z. Huang, E. Kurniawan, and C. K. Ho, "AuRoSS: An Autonomous Robotic Shelf Scanning system," in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sep. 2015, pp. 6100–6105. doi: 10.1109/IROS.2015.7354246.