# Auxilio Robotics

Propelling Assistive Healthcare into the Future

---

## Fall Project Test Plan
## Team F

---

September 20, 2023


**Team Members**
Abhinav Gupta
Atharva Pusalkar
Praveen Venkatesh
Shaolin Kataria
Shivam Tripathy

**Sponsors:** Prof Yonatan Bisk (LTI), Prof Zackory Erickson (RI)

# Contents

# 1 Introduction

This document details the tests that team Auxilio Robotics will perform during the Fall Semester of 2023. For each test, the objective, associated requirements, equipment required, elements involved, personnel in charge, test location, and verification criteria have been listed. The purpose of these tests is to verify that our system meets the performance requirements listed in Appendix A. This document also contains a schedule that highlights what milestones we plan to achieve by a set PR date, the tests it'll involve, and the requirements that will be covered.

In the previous semester, we satisfied almost all of the requirements for the Fall Validation Demo, except for manipulating an increased number of objects. Thus, we have split our tests into two parts: a) Tests required for FVD (not repeating tests done in Spring '23), and b) Tests for Desirable Requirements. This is in accordance with the project plan that has been presented during class, where we take our project in two directions, one for deployment, and another to explore experimental features that relax key constraints of the system.

Please note that we are still in active touch with our stakeholders and have not concluded on the exact specifics of the desirable requirements. However, an initial description of our idea has been included in appendix A. We have not yet defined performance requirements for the desirable goals that we will be pursuing. We have also not defined any desirable requirements for the experimental track as of the time of submitting this report, and thus have not included any tests for the same as of yet.

# 2 Logistics

All tests will be performed in our development and testing space - AI Makerspace, Tepper School of Business. For the Deployment team, their tests will be conducted at the facility. A single team member will be the primary person in charge of the tests. However, an assistant will help the lead tester with required tasks, such as keeping an eye on the robot in case of unpredictable behavior.

# 3 Schedule

Please note that the following schedule is highly tentative depending on the requirements that will be defined more concretely by the first internal milestone - IM1 on 24 September depending on stakeholder discussions.

| PR # | Date | Capability Milestone | Test | Requirements |
|---|---|---|---|---|
| PR#8 | 27 Sep | - The robot can detect and perform manipulation on 10 different object types | MRT2, MRT3, MRT4 | Appendix - Table 2 |
| PR#9 | 11 Oct | - An initial MVP system is ready for the deployment testing at the facility.<br>- Physical button based human-robot-interaction device is ready. | DDRT1, DDRT5 | Deployment desirable requirements (see section A.1) |
| PR#10 | 1 Nov | - Bug fixes are completed at the facility.<br>- Desired navigation and manipulation performance achieved. | DDRT2, DDRT3, DDRT4, DDRT6 | Deployment desirable requirements (see section A.1) |
| PR#11 | 15 Nov | - Complete FVD dry run is demonstrated | MRT1, FVD | Dry Run (as in table 1, section 4.11) |
| FVD | 20 Nov | - Entire system is demonstrated | FVD | Section 4.11 |
| FVD Encore | 29 Nov | - Entire system with desirable requirements is demonstrated (both experimental track, as well as deployment track) | TBD depending on stakeholder discussion | TBD depending on stakeholder discussion |

# 4 Tests

## 4.1 Mandatory Requirements Tests (MRT)

### 4.1.1 Test 1: HRI Task Interpretation Accuracy Test

| MRT1: HRI Task Interpretation Accuracy Test \| Objective | |
|---|---|
| Assess the accuracy of robot's understanding of a voice command and its interpretation as a task. | |
| **Associated Requirements** | M.P.1.1 (M.F.1) |
| **Equipment** | Stretch RE1, Tablet |
| **Elements** | HRI Subsystem - Speech Recognition and Task Identification |
| **Personnel** | Lead: Atharva \| Assistant: Praveen |
| **Location** | AI Makerspace (Tepper School of Business) |
| **Procedure** | |
| 1. Turn the robot on and home it.<br>2. Activate the speech recognition pipeline.<br>3. Praveen to play the role of a patient and will speak (in a normal voice) a pre-defined sentence from our test verification script (ground truth). This is a task that can either be pick-and-place, navigating or just a voice reply.<br>4. The robot acts on the voice command.<br>5. The robot metric data highlights the task being performed. This is noted as the experimental data.<br>6. The test verification script compares the task being performed (experimental data) and the desired task in the script that Praveen implied (ground truth). If the two things don't match, a flag is raised.<br>7. Steps 3-6 are repeated 15 times.<br>8. Once step 7 is finished, kill the script, and turn the robot off. | |
| **Verification Criteria** | |
| The robot was successfully able to understand the tasks that the patient (Praveen) delegated. Out of 15 attempts, the flags were less than 6, meaning that the robot was successfully able to interpret 10 speech templates as task. | |

### 4.1.2 Test 2: Object Detection Preliminary Test

| MRT2: Object Detection Preliminary Test \| Objective | |
|---|---|
| Assess the performance of the object detection pipeline for 10 object categories under controlled conditions. Tentative object list in appendix A | |
| **Associated Requirements** | M.P.6 (M.F.6) |
| **Equipment** | Stretch RE1, remote compute server, and coloured cellophane tape |
| **Elements** | Manipulation Subsystem - Perception |
| **Personnel** | Lead: Praveen \| Assitant: Abhinav |
| **Location** | AI Makerspace (Tepper School of Business) |
| **Procedure** | |
| 1. Setup the test environment as follows: a) Place the robot at a fixed location in front of the table. b) Mark 5 locations on the table (using coloured tape) where a given object can be placed. Ensure that there is sufficient lighting on the table. c) Place 5 different bottles of varying shapes and sizes on each of the locations of interest.<br>2. Turn on the robot, home it and enable the camera node.<br>3. Using the robot, collect a dataset of images using the "rosbag" tool that is supplied with ROS. The dataset should be collected as follows: a) With a given setup of 5 bottles in their locations, run the rosbag for 5 seconds. b) Interchange the positions of the bottles, and repeat "3. a)". This is done 3 times.<br>4. Using the collected dataset, mark the bounding boxes on the actual location of the object in the image on the dataset. This will serve as the ground truth.<br>5. Run the object detection algorithm on the collected rosbag, and compute the mAP with respect to the annotated ground truth collected in the previous step.<br>6. Kill the script and turn the robot off. | |
| **Verification Criteria** | |
| The computed mAP is >= 75% for different kinds of bottles in each 5 second interval. | |

### 4.1.3 Test 3: Manipulation Pick-and-Place Success Rate Test

| MRT3: Manipulation Pick-and-Place Success Rate Test \| Objective | |
|---|---|
| Assess the success rate of robot's ability to pick-and-place objects. | |
| **Associated Requirements** | M.P.7.1 (M.F.7) |
| **Equipment** | Stretch RE1, Tablet, objects for grasping |
| **Elements** | Manipulation Subsystem |
| **Personnel** | Lead: Shaolin \| Assistant: Praveen |
| **Location** | AI Makerspace (Tepper School of Business) |
| **Procedure** | |
| 1. Turn the robot on and home it.<br>2. Praveen to play the role of a patient and give a voice command that is a pick-and-place task.<br>3. The robot acts on the voice command.<br>4. Shaolin plays the role of emergency rescue and stays on standby near the robot to be able to press the kill switch button in case the robot starts behaving undesirably.<br>5. Make a note of success/ failure of the pick-and-place task.<br>6. Repeat steps 2-5 for 10 times.<br>7. Once step 6 is finished, kill the script, and turn the robot off. | |
| **Verification Criteria** | |
| The robot was able to pick-and-place objects based on Praveen's (patient) input at least 7 times. | |

### 4.1.4 Test 4: Manipulation Speed Test

| MRT4: Manipulation Speed Test \| Objective | |
|---|---|
| Assess the speed of execution of manipulation tasks. | |
| **Associated Requirements** | M.P.7.2 (M.F.7) |
| **Equipment** | Stretch RE1, objects for grasping |
| **Elements** | Manipulation Subsystem |
| **Personnel** | Lead: Shaolin \| Assistant: Praveen |
| **Location** | AI Makerspace (Tepper School of Business) |
| **Procedure** | |
| 1. Setup the test environment as follows: a) Place the robot in front of a table. b) Ensure adequate lighting is present. c) Keep 10 objects (of different categories) ready for testing.<br>2. Pick an object for manipulation and place it on the table within the field of view of the robot.<br>3. Start the robot and home it.<br>4. Issue a command to pick up the object of interest. Note down the start time (programmatically).<br>5. Once the object has been picked up, note down the time (programmatically) it took to execute said task.<br>6. Issue a command to place the object of interest. Note down the start time (programmatically).<br>7. Once the object has been placed down, note down the time (programmatically) it took to execute said task.<br>8. Repeat steps 4-7 for all 10 different objects.<br>9. Kill the script and turn the robot off. | |
| **Verification Criteria** | |
| Each manipulation task (pick-and-place individually) complete within 8 minutes of issuing the command. | |

## 4.2 Desirable Deployment Requirements Tests (DDRT)

### 4.2.1 Test 1: Command Recognition Test

| DDRT 1: Command Recognition Test \| Objective | |
|---|---|
| Assess the latency of the button-press recognition pipeline on the robot. | |
| **Associated Requirements** | M.P.1.2 (M.F.1) |
| **Equipment** | Stretch RE1, Button |
| **Elements** | HRI - Speech Recognition |
| **Personnel** | Lead: Shaolin \| Assistant: Shivam |
| **Location** | Schenley Gardens, Vincentian |
| **Procedure** | |
| 1. Turn the robot on and home it.<br>2. Press the button.<br>3. Activate the test script that measures the time between press of button and transcription.<br>4. Shivam (resident) will press the button.<br>5. The button-press will be transcribed. The time duration for this is measured using the test script.<br>6. Steps 3-5 are repeated 10 times. The mean time for transcription is then calculated.<br>7. Kill the script and turn the robot off. | |
| **Verification Criteria** | |
| Average transcription time should be less than 3.5 seconds. Although our performance requirements have a desired latency of within 5 seconds, we are only partially testing the robot's button-press system as a transcription to control module is to be implemented as well, and thus we require the robot to transcribe test within 2.5 seconds of completing the command. We will test this at a later point of time. | |

### 4.2.2 Test 2: Localization Test

| DDRT 2: Localization Test \| Objective | |
|---|---|
| Assess the performance of the localization algorithm. | |
| **Associated Requirements** | M.P.3 (M.F.3) |
| **Equipment** | Stretch RE1, coloured cellophane tape |
| **Elements** | Navigation Subsystem |
| **Personnel** | Lead: Shivam \| Assistant: Shaolin |
| **Location** | Schenley Gardens, Vincentian |
| **Procedure** | |
| 1. Ensure that the room in which the robot is navigating has been mapped and that the map has been loaded on the robot.<br>2. Select 10 different locations of interest in the testing environment. Obtain their locations (x, y) and mark the ground using coloured cellophane tape. This location is obtained using a traditional tape measure with respect to other local objects near the location of interest.<br>3. Start the robot and home it.<br>4. Enable the localization algorithm node on the robot.<br>5. From an arbitrary location (that is not the location of interest), teleoperate the robot to each of the locations of interest sequentially, pausing at each location for a duration of 10 seconds and noting down the estimated localization coordinates.<br>6. Kill the script and turn the robot off. | |
| **Verification Criteria** | |
| The average L2 norm between the true and estimated localization coordinates should be less than 25cms. | |

### 4.2.3 Test 3: Manipulation Speed Test

| DDRT 3: Manipulation Speed Test \| Objective | |
|---|---|
| Assess the speed of execution of manipulation tasks. | |
| **Associated Requ** | M.P.7.2 (M.F.7) |
| **Equipment** | Stretch RE1, objects for grasping |
| **Elements** | Manipulation Subsystem |
| **Personnel** | Lead: Shaolin \| Assistant: Shivam |
| **Location** | Schenley Gardens, Vincentian |
| **Procedure** | |
| 1. Setup the test environment as follows: a) Place the robot in front of a common table. b) Ensure adequate lighting is present. c) Keep 3 objects (of different categories) ready for testing.<br>2. Pick an object for manipulation and place it on the table within the field of view of the robot.<br>3. Start the robot and home it.<br>4. Issue a command to pick up the object of interest. Note down the start time (programmatically).<br>5. Once the object has been picked up, note down the time (programmatically) it took to execute said task.<br>6. Issue a command to place the object of interest. Note down the start time (programmatically).<br>7. Once the object has been placed down, note down the time (programmatically) it took to execute said task.<br>8. Repeat steps 4-7 for all 10 different objects.<br>9. Kill the script and turn the robot off. | |
| **Verification Criteria** | |
| Each manipulation task (pick-and-place individually) complete within 8 minutes of issuing the command. | |

### 4.2.4 Test 4: Obstacle Avoidance Test

| DDRT 4: Obstacle Avoidance Test \| Objective | |
|---|---|
| Assess obstacle avoidance against 10 static obstacles. | |
| **Associated Requirements** | M.P.5 (M.F.5) |
| **Equipment** | Stretch RE1, 10 static obstacles |
| **Elements** | Navigation Subsystem |
| **Personnel** | Lead: Shivam \| Assistant: Shaolin |
| **Location** | Schenley Gardens, Vincentian |
| **Procedure** | |
| 1. Turn the robot on and home it. Check if the robotic arm is stowed.<br>2. If the arm is not stowed, run the script to stow the arm.<br>3. Set up a test environment with 10 different obstacles (with varying shapes and sizes) on a path from the robot's starting location to the robot's goal location.<br>4. Issue a navigation command for the robot from the start to the end location.<br>5. Abhinav to stay on-alert within the vicinity of the robot to be able to press the kill switch button in case the robot starts behaving undesirably.<br>6. Make a note of success/ failure as the robot passes a pre-defined list of 10 objects.<br>7. Kill the script and turn the robot off. | |
| **Verification Criteria** | |
| The robot was able to navigate in the desired path while avoiding all 10 obstacles. | |

### 4.2.5 Test 5: Teleoperation Command Latency Test

| DDRT 5: Teleoperation Command Latency Test \| Objective | |
|---|---|
| Assess latency of teleoperation commands on the robot for 3 minutes. | |
| **Associated Requirements** | M.P.8 (M.F.8) |
| **Equipment** | Stretch RE1, Tablet |
| **Elements** | HRI Subsystem - Teleoperation |
| **Personnel** | Lead: Shaolin \| Assistant: Shivam |
| **Location** | Schenley Gardens, Vincentian |
| **Procedure** | |
| 1. Turn the robot on and home it.<br>2. Shivam to press the button.<br>3. Shaolin to play the role of a loved one and teleoperate the robot while connecting to the video call with Shivam.<br>4. Shaolin gives commands on the application on the tablet to maneuever the robot. This is noted as timestamp 1 in our test verification script.<br>5. Shivam now plays the role of emergency rescue and stays on standby near the robot to be able to press the kill switch button in case the robot starts behaving undesirably.<br>6. Our test verification script notes the second timestamp which depicts the time it takes for the robot to execute the command that Shaolin passed as the instruction on the application on the tablet. This can be either change of coordinates in navigation, or change of pose of arm in manipulation.<br>7. Repeat steps 4-6 for 3 minutes and store the timestamp differences as an array.<br>8. Once step 7 is finished, disconnect the user from the teleoperation mode, kill the script, and turn the robot off. | |
| **Verification Criteria** | |
| The loved one (Shaolin) was able to teleoperate the robot with the time between his commands and the resulting robot operation (latency) not once exceeding 5 seconds. The proof of this will be visible in the timestamp difference array for a period of 3 minutes. | |

### 4.2.6 Test 6: Manipulation Pick-and-Place Success Rate Test

| DDRT 6: Manipulation Pick-and-Place Success Rate Test \| Objective | |
|---|---|
| Assess the success rate of robot's ability to pick-and-place objects. | |
| **Associated Requ** | M.P.7.1 (M.F.7) |
| **Equipment** | Stretch RE1, Tablet, objects for grasping |
| **Elements** | Manipulation Subsystem |
| **Personnel** | Lead: Shaolin \| Assistant: Shivam |
| **Location** | Schenley Gardens, Vincentian |
| **Procedure** | |
| 1. Turn the robot on and home it.<br>2. Shaolin to play the role of a patient and give a voice command that is a pick-and-place task.<br>3. The robot acts on the voice command.<br>4. Shivam plays the role of emergency rescue and stays on standby near the robot to be able to press the kill switch button in case the robot starts behaving undesirably.<br>5. Make a note of success/ failure of the pick-and-place task.<br>6. Repeat steps 2-5 for 10 times.<br>7. Once step 6 is finished, kill the script, and turn the robot off. | |
| **Verification Criteria** | |
| The robot was able to pick-and-place objects based on Shaolin's (patient) input at least 4 times. | |

## 4.3 Test FVD: Fall Validation Demo (Experiment)

- **Associated Requirements**: As listed on table 1
- **Equipment**: Stretch RE1, Objects for Manipulation
- **Elements**: Manipulation, Navigation, Human-Robot Interaction
- **Personnel**: Abhinav, Atharva, Praveen, Shaolin, Shivam
- **Location**: AI Makerspace (Tepper School of Business)
- **Environment setup**: We set up the environment in the following way:
    1. Ten different objects are placed on a table that is placed in a location that is not near the start location of the robot.
    2. Place obstacles in the path between the object and the user's location.

**Table 1: Fall Demo Procedure**

| Steps | Procedure | Success Criteria | Requirements |
|---|---|---|---|
| **1** | Pam (senior citizen) asks Alfred to set up a video call with his/her family | Successful command interpretation and video calling | **- M.P.1 (partial)**<br>**- M.N.3** |
| **2** | User (Pam's family members) teleoperates the robot using the app in handheld device to pick up an object for Pam. | Communication Latency <5s (visual verification) | **M.P.8** |
| **3** | Robot prevents obstacles while moving in teleoperation and performs manipulation. | Avoids 100% of obstacles | **M.P.5** |
| **4** | Pam gives speech commands to Alfred to pick up an object. | Correctly interprets speech with communication latency <5s (visual verification) | **M.P.1 (partial)** |
| **5** | Robot localizes itself in the premapped environment | Localization with error threshold<25 cms | **M.P.3** |
| **6** | Robot interprets the command and plans a global path using predefined heuristics | - Correct interpretation of the task<br>- Plans global path within 2 minutes | **M.P.1.1, M.P.4.1** |
| **7** | Robot navigates and reaches the goal location while avoiding static and dynamic obstacles. | - Avoids 100% of obstacles<br>- Navigates at an average speed of 0.4 m/s | **M.P.5, M.P.4.2** |

| | | | |
|---|---|---|---|
| **8** | Robot detects the object and estimates the grasping location | Successful object detection for chosen object category. | **M.P.6 (partial)** |
| **9** | The robot picks up the object. | - Successfully picks up the object within 3 tries.<br>- Time to completion <=8min | **M.P.7.1, M.P.7.2** |
| **10** | Robot navigates back to starting location and places the object on the user's table. | - Avoids 100% of obstacles<br>- Navigates at an average speed of 0.4 m/s<br>- Greater than 70% successful placements | **M.P.5, M.P.4.2** |
| **11** | Robot metrics and video feed of the robot are displayed on device of the user throughout the mission | Communication Latency <5s (visual verification) | **M.P.9** |

# A Appendix

**Table 2: Mandatory Functional and Performance Requirements**

| Functional | Performance | Description |
|---|---|---|
| **M.F.1** Receive commands from the user: preset speech primitives/handheld interface | **M.P.1.1** Interpret 10 speech templates as tasks.<br>**M.P.1.2** Latency for control commands <5s | The robot will primarily operate using speech inputs. This should be seamless for the user. |
| **M.F.2** Perform basic (predefined) social engagement with user | **M.P.2** Fallback rate: <20% | Robot provides feedback upon receiving commands, and should be able to automatically execute pre-defined tasks. |
| **M.F.3** Localize itself in the environment | **M.P.3** Average error <25 cms | |
| **M.F.4** Plan and navigate through the pre-mapped environment | **M.P.4.1** Plan global path to desired location within 2 minutes.<br>**M.P.4.2** Navigate at a speed of 0.4 m/s | |
| **M.F.5** Autonomously avoid obstacles in the environment | **M.P.5** Avoid 100% of static obstacles in range | The robot should not collide with any object as it may render the environment unsafe for users. |
| **M.F.6** Detect objects for grasping | **M.P.6** mAP >= 80% for 10 object categories (e.g bottle, remote, medicines etc) under the following conditions: 1) Object is within 1m of body camera 2) Not kept on a white/transparent surface 3) Adequate Indoor lighting conditions | The conditions have been prespecified keeping in mind that a robust system in a structured environment is more preferred than an unreliable system in a general environment |
| **M.F.7** Manipulate predefined objects to/from planar surfaces at known locations in the environment | **M.P.7.1** Greater than 70% successful picks and places<br>**M.P.7.2** Manipulation tasks should be completed within 8 min. | The objects will be picked up and placed on flat planar surfaces such as tables only. i.e, allowing "the user to grasp and remove the object from the end-effector" is out of scope. |
| **M.F.8** Allow approved operators to teleoperate the robot | **M.P.8** Communication latency <5s | |
| **M.F.9** Provide user with robot metrics and video feed of the robot on a handheld interface | **M.P.9** Communication latency: <2s | |

**Table 3: Mandatory Non Functional Requirements**

| Requirements | |
|---|---|
| **M.N.1** | Appear non-threatening to the user |
| **M.N.2** | Be physically compliant to human interaction/contact |
| **M.N.3** | Have a simple UI/UX for the handheld interface |
| **M.N.4** | Have a modular software architecture for further development |
| **M.N.5** | Allow users to pre-schedule tasks/assistance |

**Table 4: Desirable Non-Functional Requirements**

| Requirements | |
|---|---|
| **D.N.1** | Appear aesthetic to the user |
| **D.N.2** | Reasonable cost for the user |

## A.1 Tentative Description of System to be Deployed

We envision the robot that is to be deployed to perform autonomous pick and place in an assisted living facility. This is similar to what we demonstrated in the spring validation demo, albeit with a few key changes. They are tentatively as follows:

- Replace the voice-based input system of the robot with a simple wireless physical button-remote system for ease of use by the elderly.
- The perception stack for placing/picking will be simplified and will make extensive use of fiducial markers for improving the robustness of tasks carried out by the system.

Kindly note that this is still tentative, and is prone to change depending on discussions with stakeholders.