

User Manual

A GUI for controlling and supervising
multiple robots remotely

Version 1.0
30th November 2019

Prepared by

- Jaliyagoda A.J.N.M.
- Tennakoon T.M.P.B.
- Karunarathna S.D.D.D

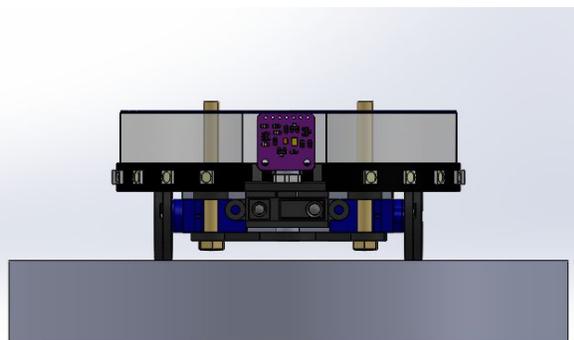
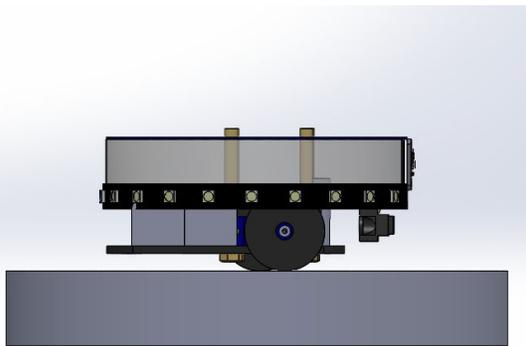
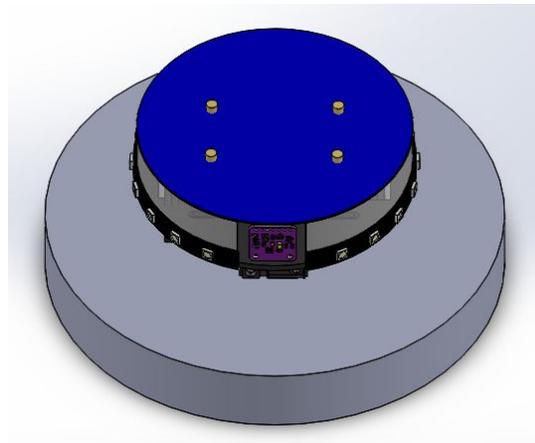
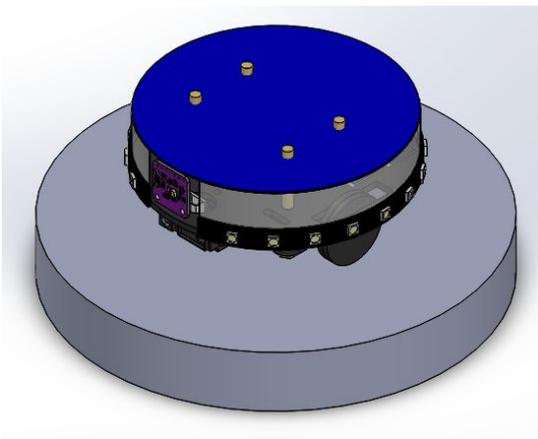
Table of Contents

General Information	2
Hardware Overview	2
Physical Structural Design	2
PCB Design	3
Schematics Design	5
Software Overview	7
REST APIs available on the robot	8
User Portal	9
Web Interface	9
API Interface	23
Scan and identify available robots	23
Obtaining the proximity reading of a robot	24
Send the moving path to a robot	25
Update the color of a robot	26

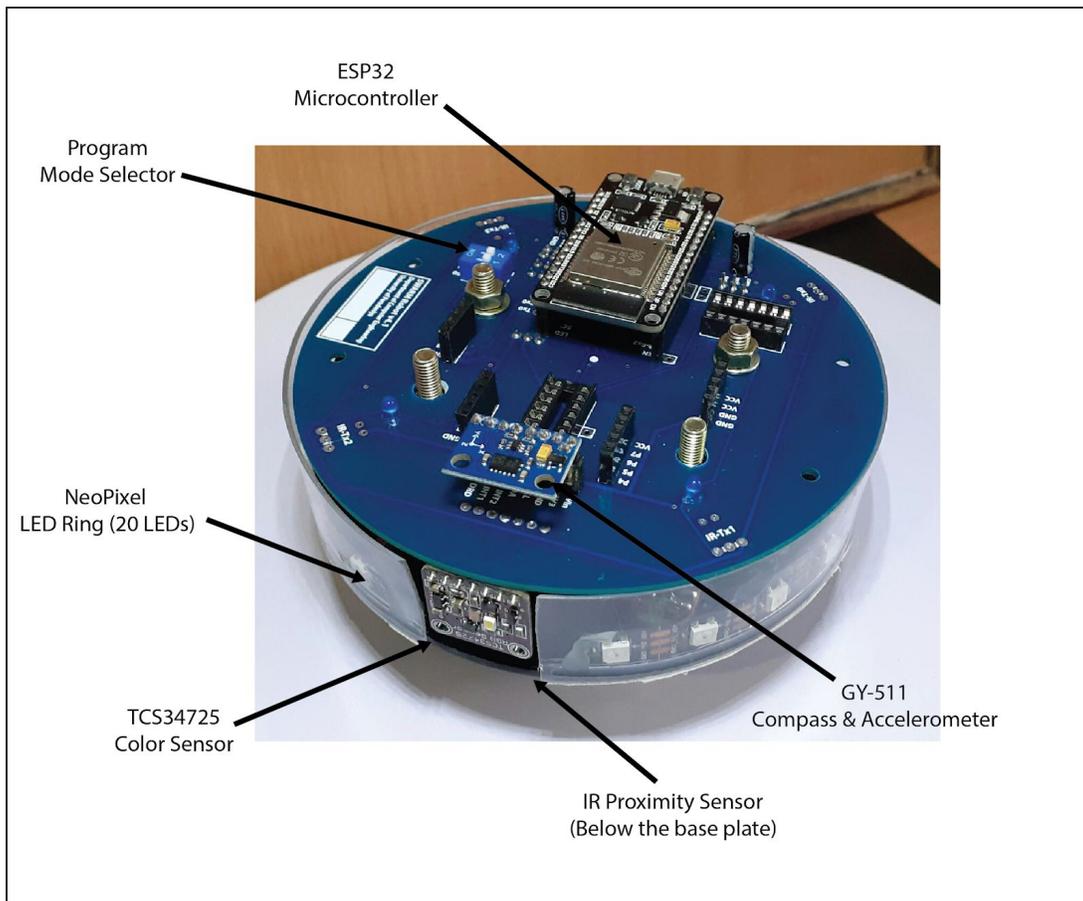
General Information

Hardware Overview

Physical Structural Design



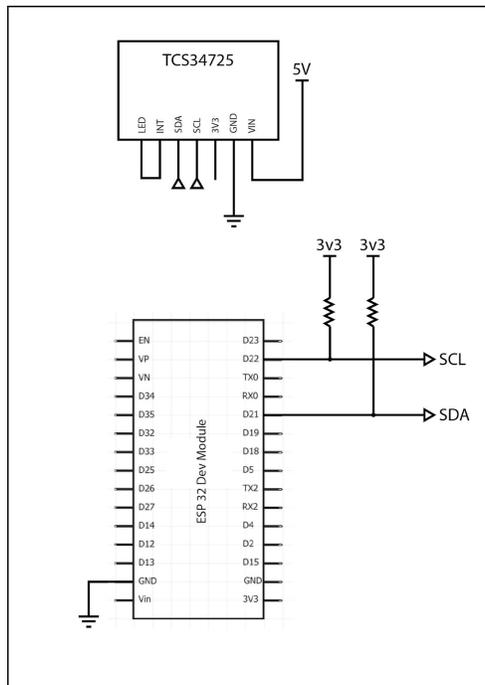
Bottom View of the PCB Layout Design



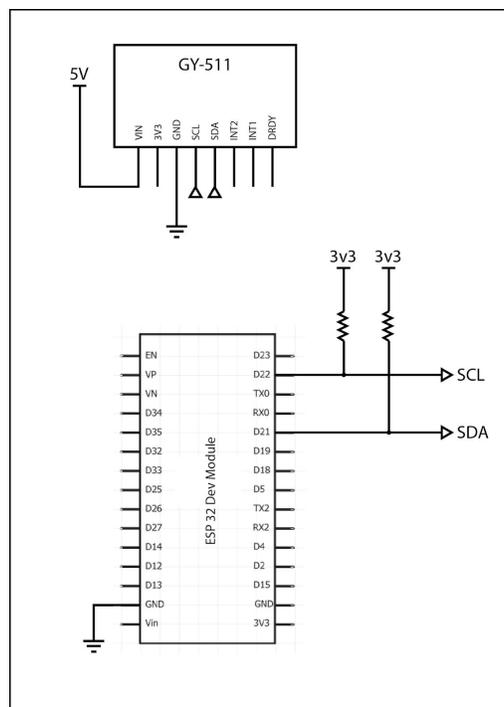
Few main component placements of the robot

Schematics Design

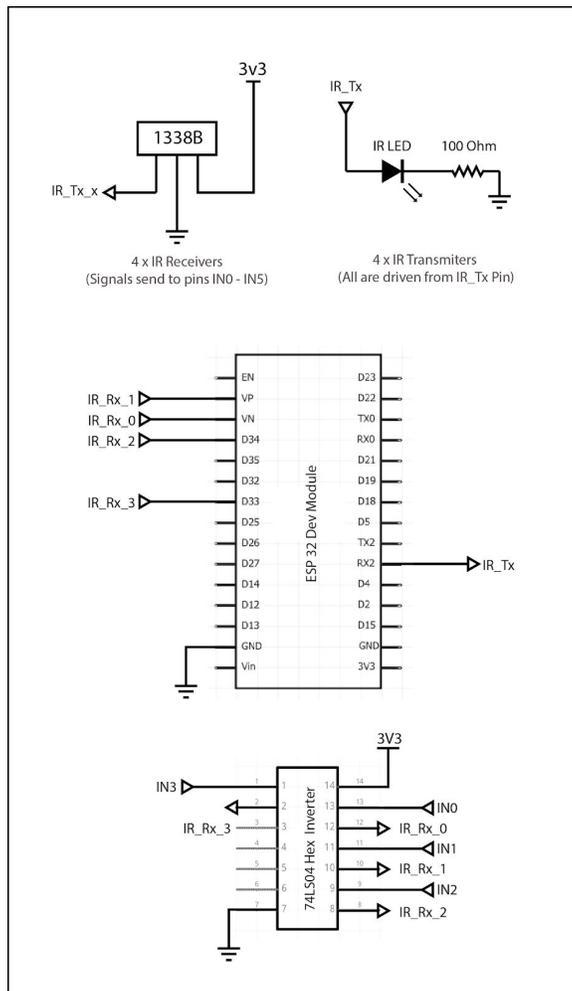
1. TCS34725 Color Sensor



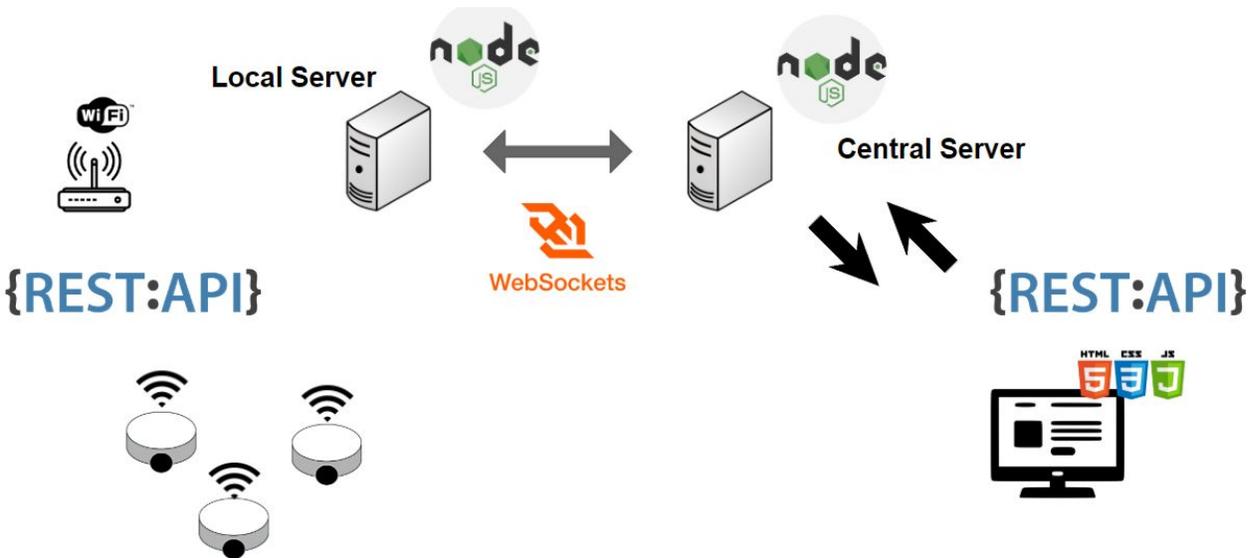
2. GY-511 6-DOF Compass and Accelerometer Sensor



3. IR Transmitter and Receiver Units



Software Overview



There are two types of servers available in communication flow. One is known as the local server, which communicates between Robot Units and the central server. This is hosted near the simulation arena and have access to hardware behaviours of the arena too (Ex: Control the environment conditions of the arena)

Other server, the central server is hosted somewhere on the internet and can be accessed by anyone using the IP address or domain of it. It can handle more than one local servers. The central server hosts the GUI (web site) and the web socket server, which communicates with local servers. Central server keeps a database, which contains configuration data and user data.

Central server and Local servers will communicate with each other using a Web Socket connection. When a local server establishing a Web socket connection with Central Server, it should provide an authentication token. The authentication token can be obtained by making HTTP Get request into central server's API gateway by providing the previous access token.

REST APIs available on the robot

Local server can identify the robots which are online on its subnet, and communicate with them using REST APIs listed below.

1. Status of the robot

`http://{Robot IP}/status`

Response: {“status”:”, “version”:”, “IP”:”}

2. Magnetometer Readings

`http://{Robot IP}/mag`

Response: {“x”:”, “y”:”, “z”:”, “heading”:”}

3. Accelerometer Readings

`http://{Robot IP}/accel`

Response: {“x”:”, “y”:”, “z”:”}

4. IR Proximity Readings

`http://{Robot IP}/dist`

Response: {“raw”:”, “filtered”:”}

5. Color sensor readings

`http://{Robot IP}/color`

Response: {“R”:”, “G”:”, “B”:”, “temp”:”, “lux”:”, }

6. Control motion of the robot

`http://{Robot IP}/motor?dir={DIR}`

DIR = [forward, backward, rotCW, rotCCW, stop]

Response: { “status”: ” }

7. Control the rotation of the robot

`http://{Robot IP}/turn?ang={Angle}`

Angle = [-180, 180]

Response: { “status”: ” }

- Control the color of the LED ring of the robot

`http://{Robot IP}/pixelLED/all?R={$1}&G={$2}&B={$3}`

\$1 = Red Value [0, 255]

\$2 = Green Value [0, 255]

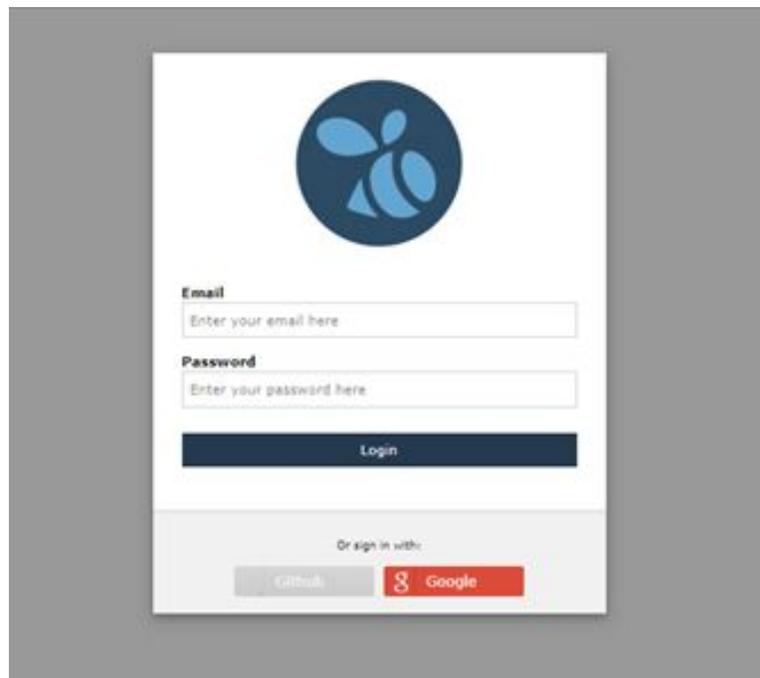
\$3 = Blue Value [0, 255]

Response: { "status": "" }

User Portal

Web Interface

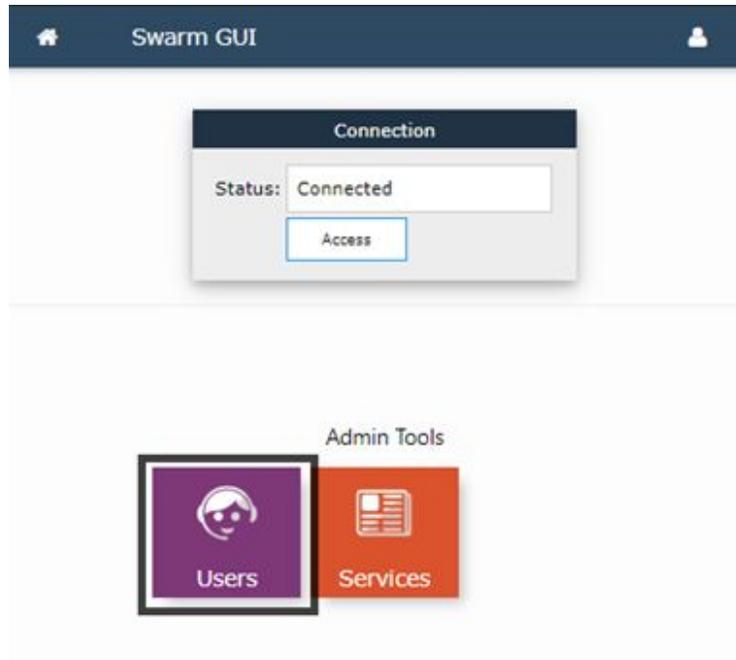
- Login window



Users can login to the system using their email address and password. They can also use sign in with google feature for the authentication This feature simplifies the login process for users and realize a higher conversion rate for registration.

2. Landing Page

After login to the system user will be able to see the following home screen. But only an administrative person can view the section, 'Admin Tools'.



2.1 Users

When user selects '**Users**' link below screen will appear. This feature is only available for administrative users and can be used to manage users.

User ID	User	Role	Last Accessed Time	Actions
100000	A.J.N.M. Jaliyagoda (nuwanjaliyagoda@eng.pdn.ac.lk)	Admin	2019-11-29 06:11:34	Edit Delete
100001	T.M.R.B Tennakoon (pasan90tennakoon@gmail.com)	Admin	2019-10-23 14:30:09	Edit Delete
100002	S.D.D.D. Karunaratna (dinelkadilshani95@gmail.com)	Admin	2019-12-01 11:51:39	Edit Delete
100010	R.G. Ragel (rosharr@ce.pdn.ac.lk)	Lecturer	2019-11-28 01:39:03	Edit Delete
100011	I. Nawinne (isurunawinne@gmail.com)	Lecturer	2019-01-03 09:01:50	Edit Delete
100026	Test User (ceykod@gmail.com)	GSuitUser	2019-11-04 23:15:59	Edit Delete
100028	Nuwan Madushanka (ncreationsrilanka@gmail.com)	GSuitUser	2019-11-28 01:14:22	Edit Delete

When user selects 'Edit' link below screen will appear.

Swarm GUI

Home / Users / Edit Users

Edit User

Salutation
Mr.

First Name (with initials)
A.J.N.M.

Last Name
Jaliyağoda

Email
nuwanjaliyagoda@eng.pdn.ac.lk

Role
Admin

Update User

When user selects 'Add new user' link in the Users window,

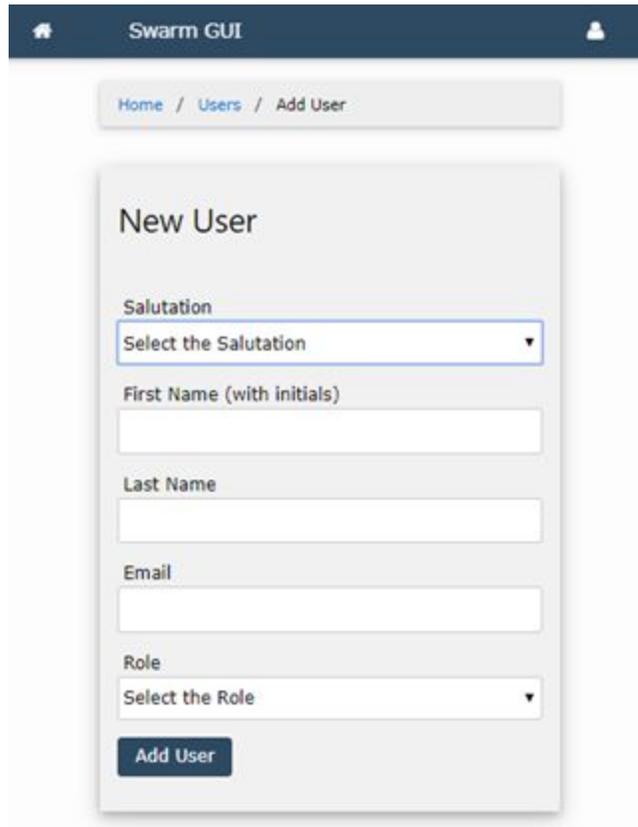
Swarm GUI

Home / Users

Portal Users Add New User

User ID	User	Role	Last Accessed Time	Actions
100000	A.J.N.M. Jaliyagoda (nuwanjaliyagoda@eng.pdn.ac.lk)	Admin	2019-11-29 06:11:34	Edit Delete
100001	T.M.P.B Tennakoon (pasan96tennakoon@gmail.com)	Admin	2019-10-23 14:30:09	Edit Delete
100002	S.D.D.D. Karunarithna (dinelkadilshani95@gmail.com)	Admin	2019-12-01 11:51:39	Edit Delete
100010	R.G. Ragel (roshanr@ce.pdn.ac.lk)	Lecturer	2019-11-28 01:39:03	Edit Delete
100011	I. Nawinne (isurunawinne@gmail.com)	Lecturer	2019-01-03 09:01:50	Edit Delete
100026	Test User (ceykod@gmail.com)	GSuitUser	2019-11-04 23:15:59	Edit Delete
100028	Nuwan Madushanka (ncreationsrilanka@gmail.com)	GSuitUser	2019-11-28 01:14:22	Edit Delete

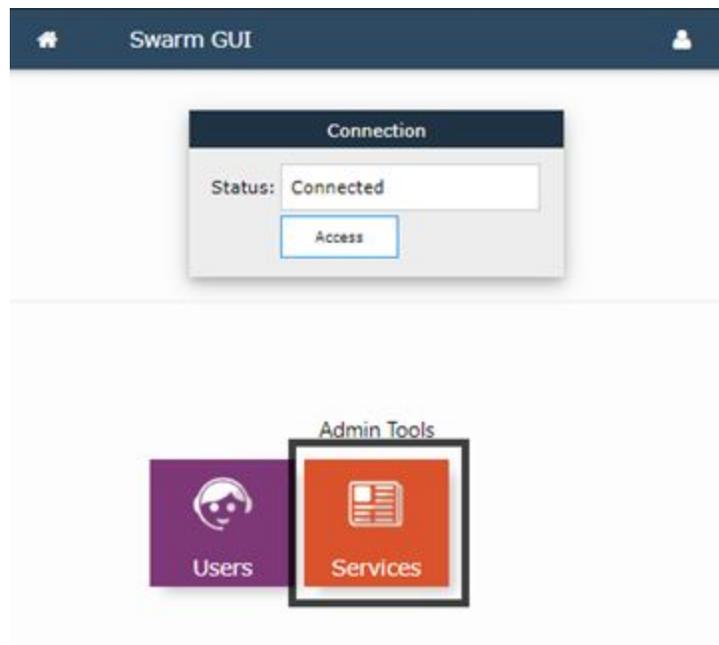
below screen will appear. So users can add new users to the system by this window



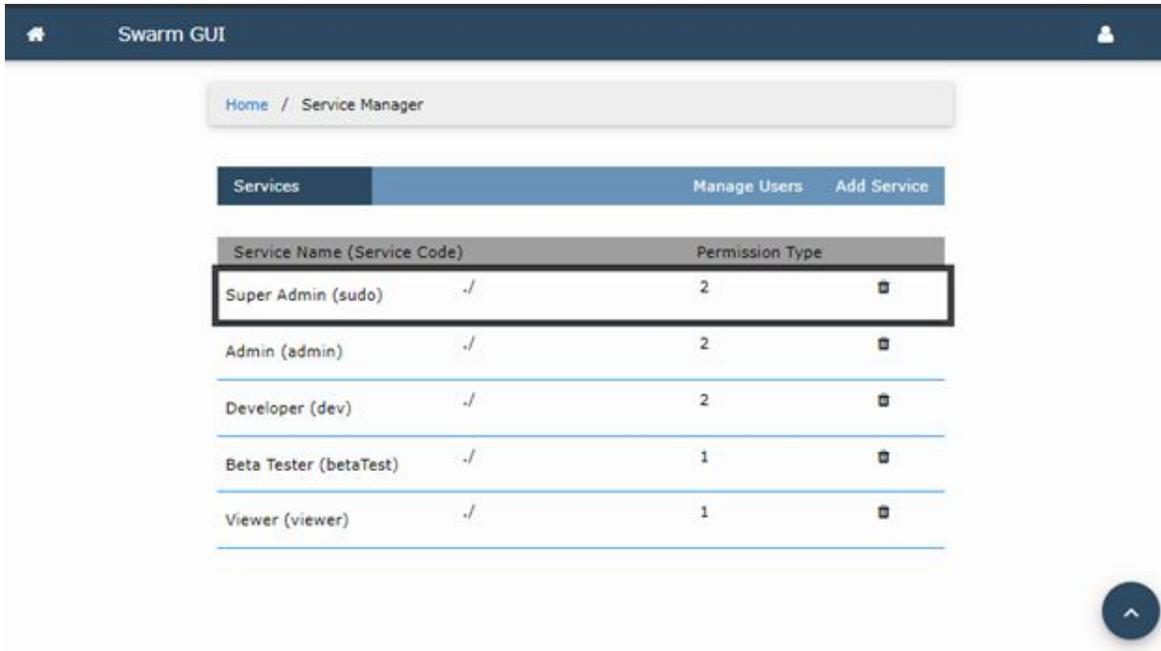
The screenshot shows the 'Swarm GUI' interface with a breadcrumb trail 'Home / Users / Add User'. The main content area is titled 'New User' and contains the following fields:

- Salutation: A dropdown menu with the text 'Select the Salutation' and a downward arrow.
- First Name (with initials): A text input field.
- Last Name: A text input field.
- Email: A text input field.
- Role: A dropdown menu with the text 'Select the Role' and a downward arrow.
- At the bottom of the form is a dark blue button labeled 'Add User'.

2.2 Services



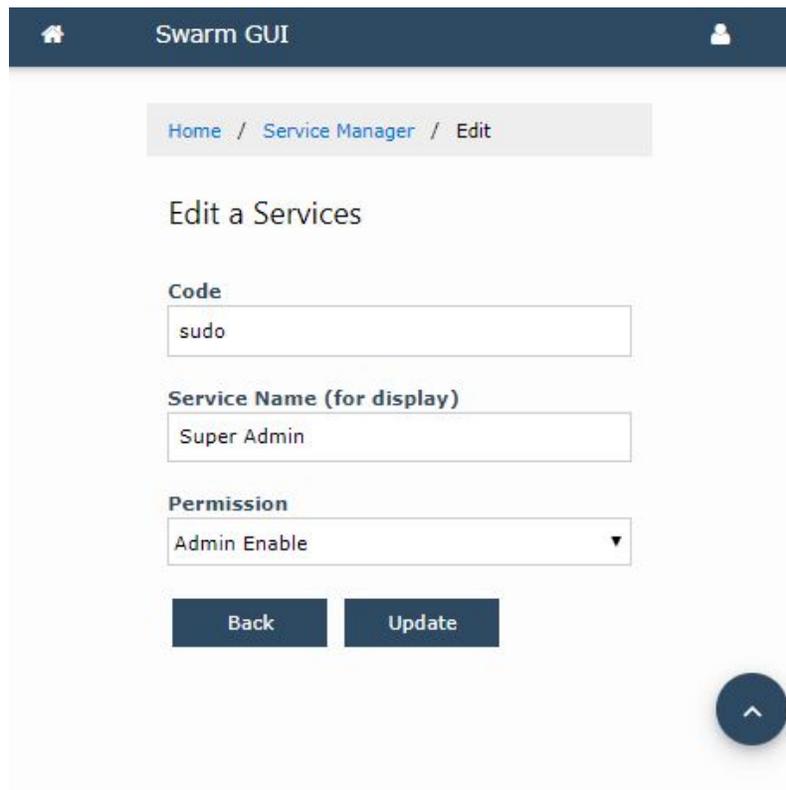
When user selects 'Services' link in the home page below screen will appear. This control the access to the system like an Access Control List. There are few categories with different access types as below,



The screenshot shows the Swarm GUI interface. At the top, there is a dark blue header with a home icon, the text "Swarm GUI", and a user profile icon. Below the header is a breadcrumb trail: "Home / Service Manager". A navigation bar contains three items: "Services" (highlighted in blue), "Manage Users", and "Add Service". Below this is a table with two columns: "Service Name (Service Code)" and "Permission Type". The table lists five service categories, with the first one, "Super Admin (sudo)", highlighted with a black border. Each row also has a small square icon on the right side.

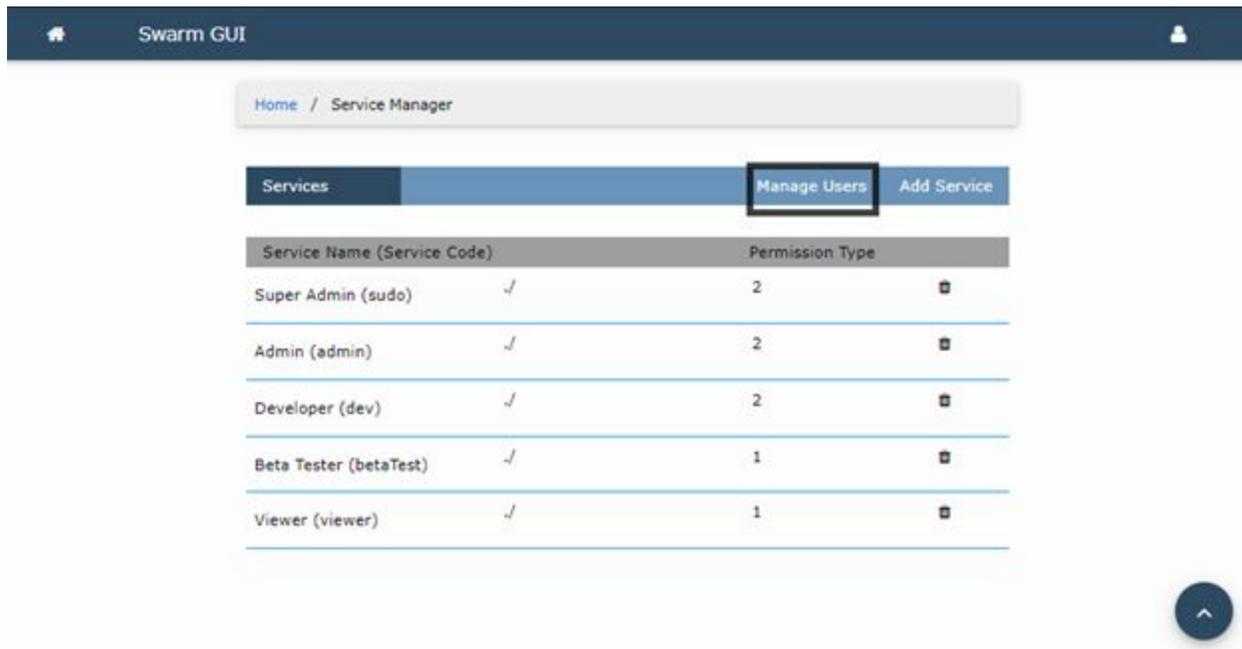
Service Name (Service Code)	Permission Type
Super Admin (sudo)	2
Admin (admin)	2
Developer (dev)	2
Beta Tester (betaTest)	1
Viewer (viewer)	1

Each category can configure like this



The screenshot shows the "Edit a Services" page in the Swarm GUI. The header is the same as the previous screenshot. The breadcrumb trail is "Home / Service Manager / Edit". The main heading is "Edit a Services". There are three form fields: "Code" with the value "sudo", "Service Name (for display)" with the value "Super Admin", and "Permission" with a dropdown menu showing "Admin Enable". At the bottom, there are two buttons: "Back" and "Update".

Select '**Manage Users**' can be used to view the status, account type and last login of each user.

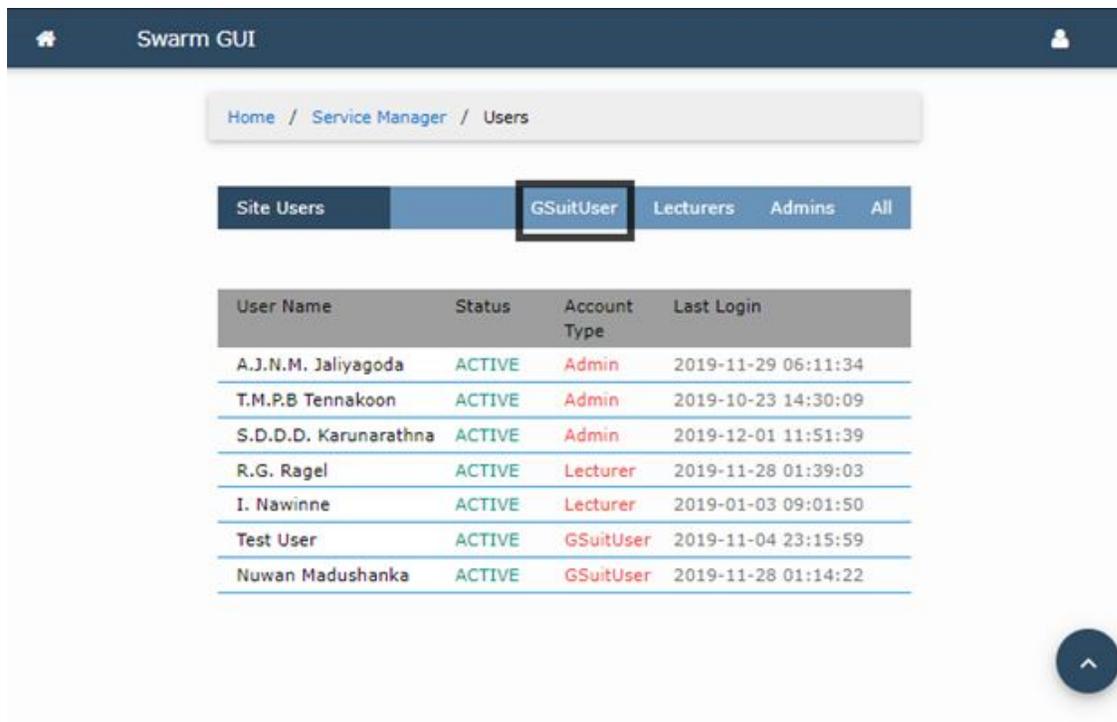


The screenshot shows the Swarm GUI interface. At the top, there is a dark blue header with a home icon and the text "Swarm GUI". Below the header is a breadcrumb trail: "Home / Service Manager". A navigation bar contains three buttons: "Services", "Manage Users" (which is highlighted with a black box), and "Add Service". Below the navigation bar is a table with the following data:

Service Name (Service Code)		Permission Type	
Super Admin (sudo)	✓	2	⊞
Admin (admin)	✓	2	⊞
Developer (dev)	✓	2	⊞
Beta Tester (betaTest)	✓	1	⊞
Viewer (viewer)	✓	1	⊞

At the bottom right of the page, there is a circular button with an upward-pointing arrow.

And also it can filter the users according to the categories they belong by selecting the particular category in the right corner

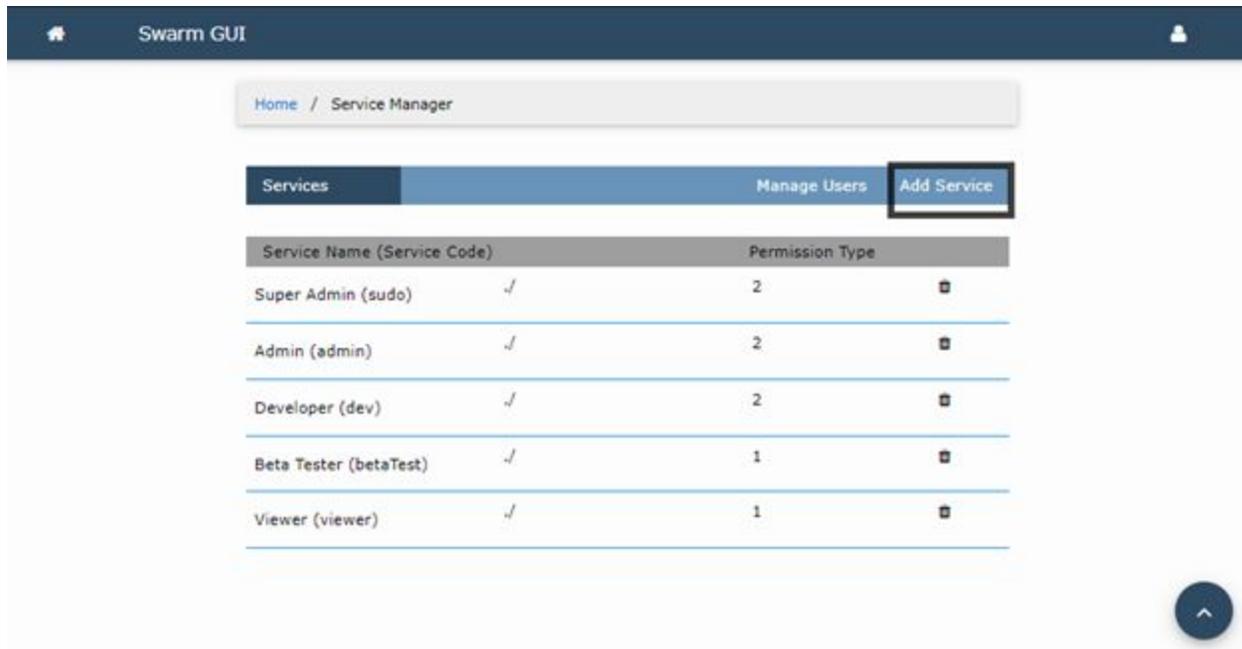


The screenshot shows the Swarm GUI interface. At the top, there is a dark blue header with a home icon and the text "Swarm GUI". Below the header is a breadcrumb trail: "Home / Service Manager / Users". A navigation bar contains five buttons: "Site Users", "GSuitUser" (which is highlighted with a black box), "Lecturers", "Admins", and "All". Below the navigation bar is a table with the following data:

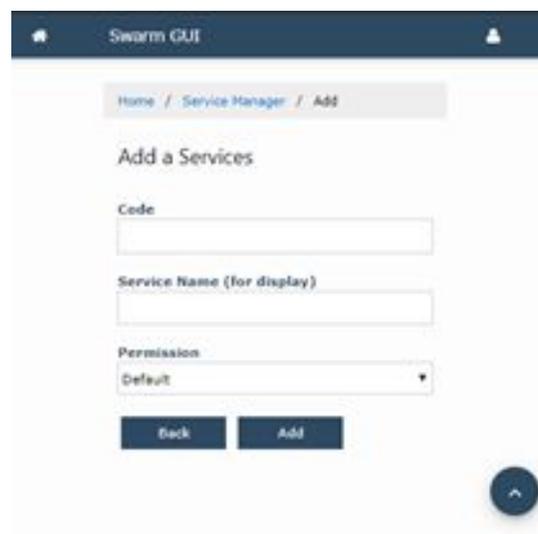
User Name	Status	Account Type	Last Login
A.J.N.M. Jaliyagoda	ACTIVE	Admin	2019-11-29 06:11:34
T.M.P.B Tennakoon	ACTIVE	Admin	2019-10-23 14:30:09
S.D.D.D. Karunarathna	ACTIVE	Admin	2019-12-01 11:51:39
R.G. Ragel	ACTIVE	Lecturer	2019-11-28 01:39:03
I. Nawinne	ACTIVE	Lecturer	2019-01-03 09:01:50
Test User	ACTIVE	GSuitUser	2019-11-04 23:15:59
Nuwan Madushanka	ACTIVE	GSuitUser	2019-11-28 01:14:22

At the bottom right of the page, there is a circular button with an upward-pointing arrow.

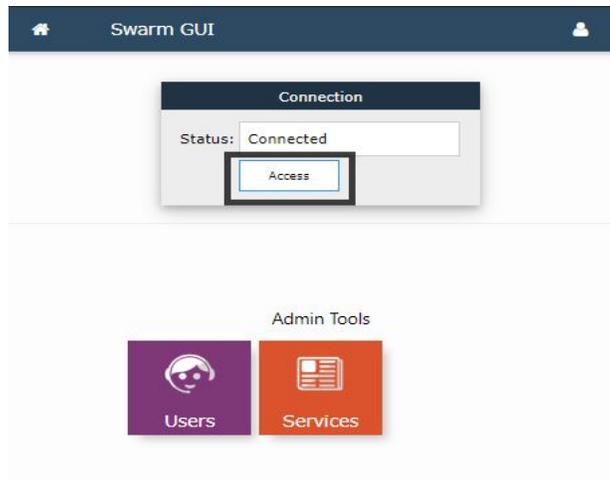
To add new services, the link **'Add services'** in Services Manager window can be used.



When user selects 'Add Services' link below screen will appear.

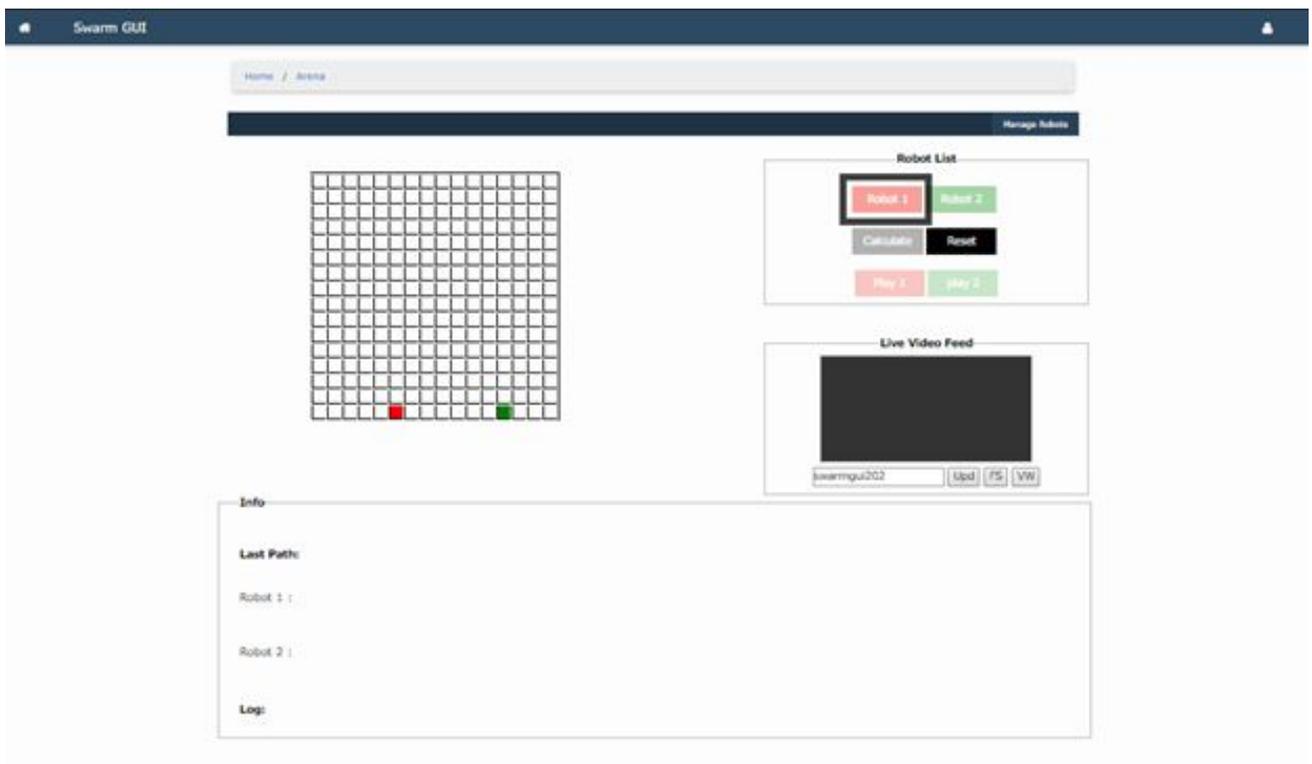


2.3 Connection

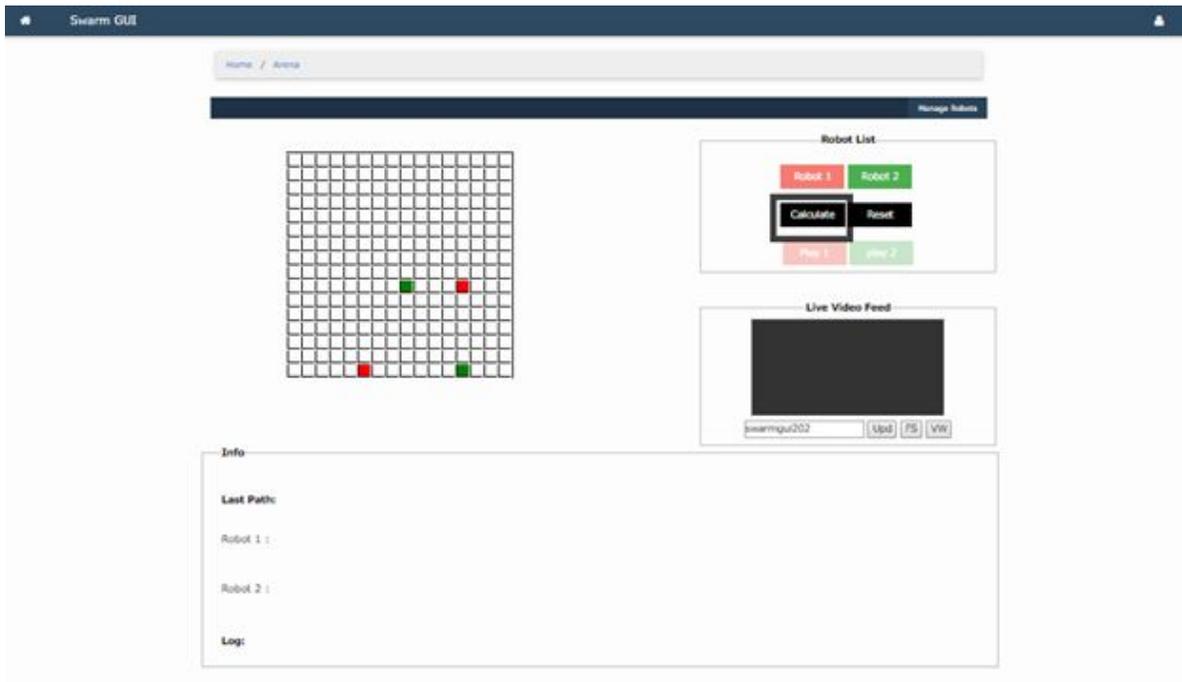
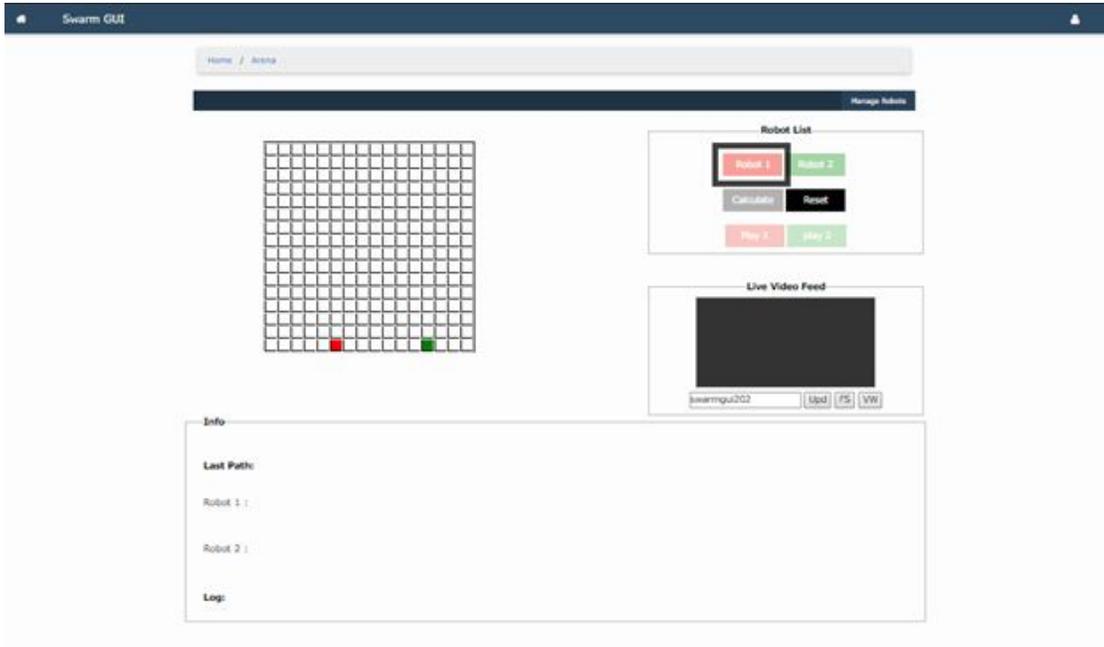


This Display the connectivity of the platform. If it is connected user can selects '**Access**' link. After that the following window will appear

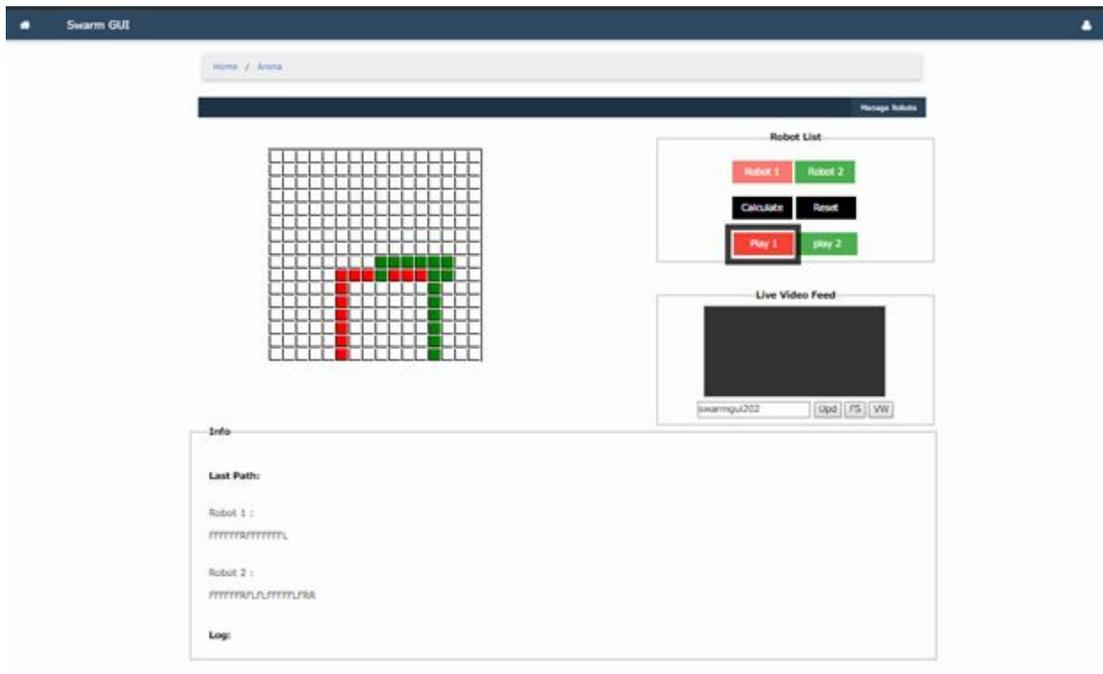
3 Grid page



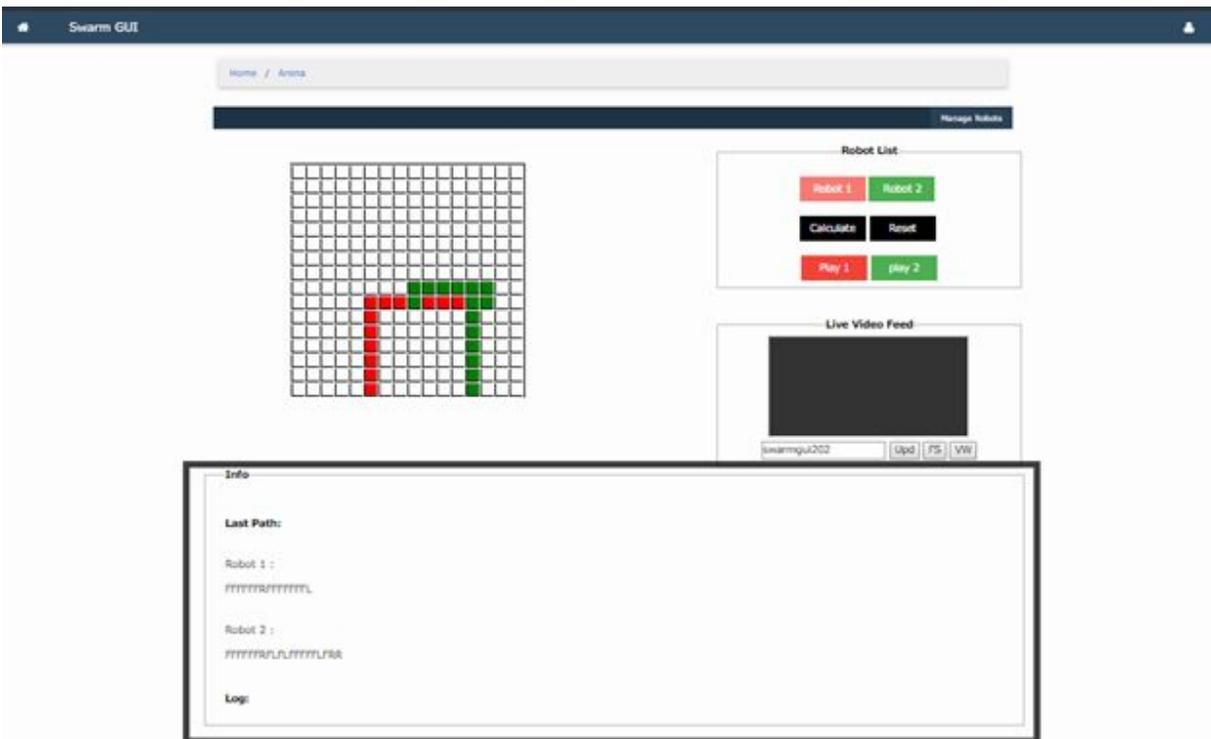
Robots should be placed in specific locations and they will be denoted in red and green color in the grid (Only two robots yet). Users can select the destinations for the robots as in the figure by clicking the particular robot and the destination point.

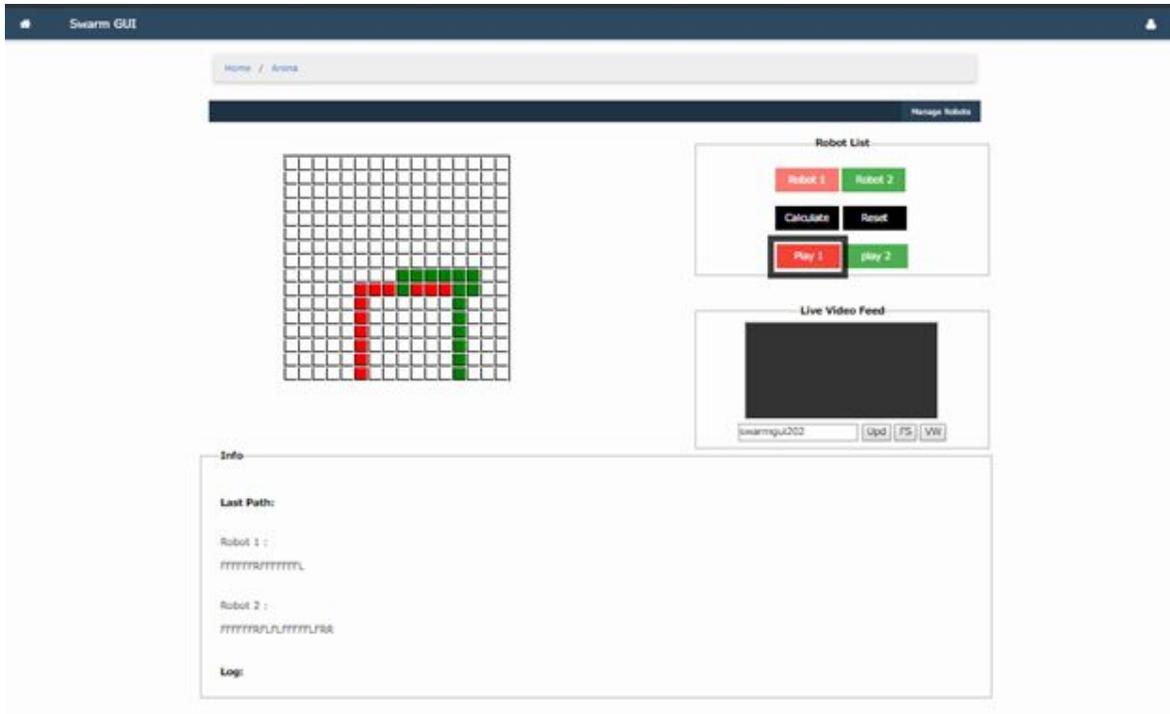


Then by clicking the **Calculate** button the system calculates the path for the robots by using a collision detect & prevent algorithm and it will be displayed in the grid as follows.

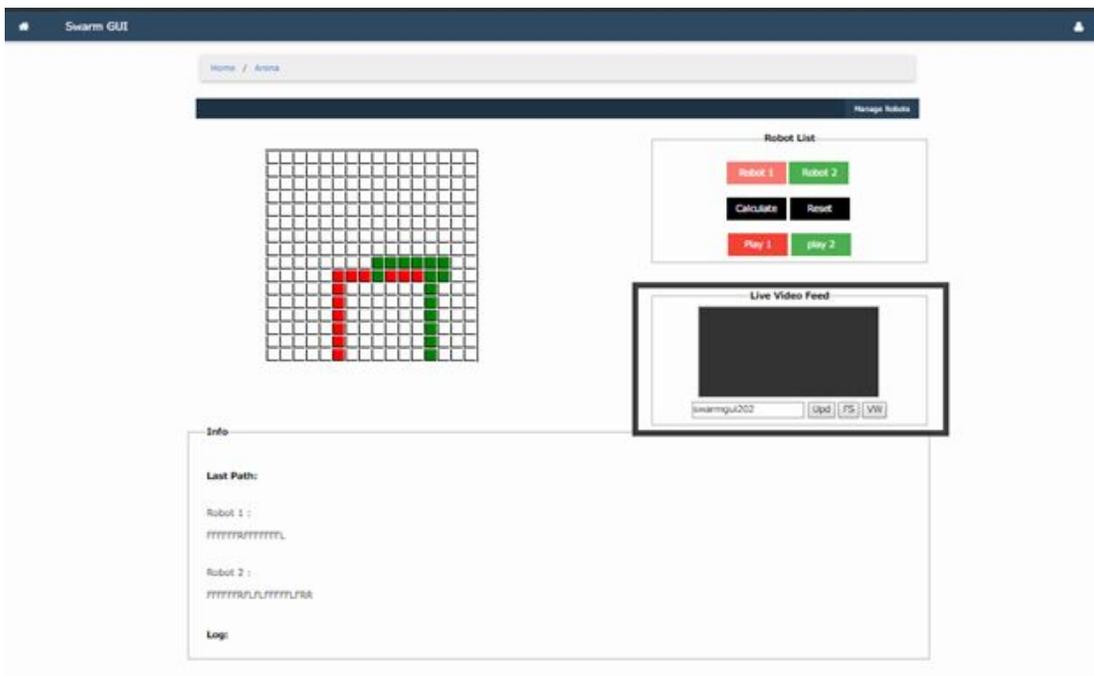


The info panel gives the path of each robot separately by the symbols 'F', 'L' and 'R' to indicate a forward, left and right respectively.





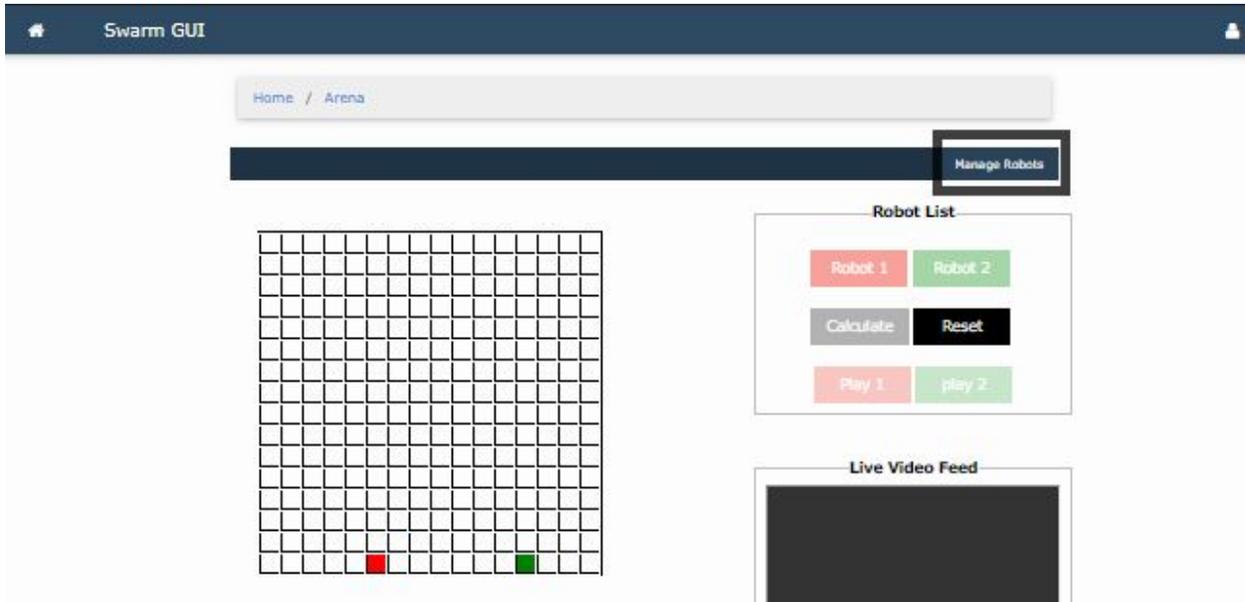
By clicking the 'Play 1' button it sends the path that calculate to the particular robot and the robot will move accordingly. 'Play 2' button does the same thing to the second robot.



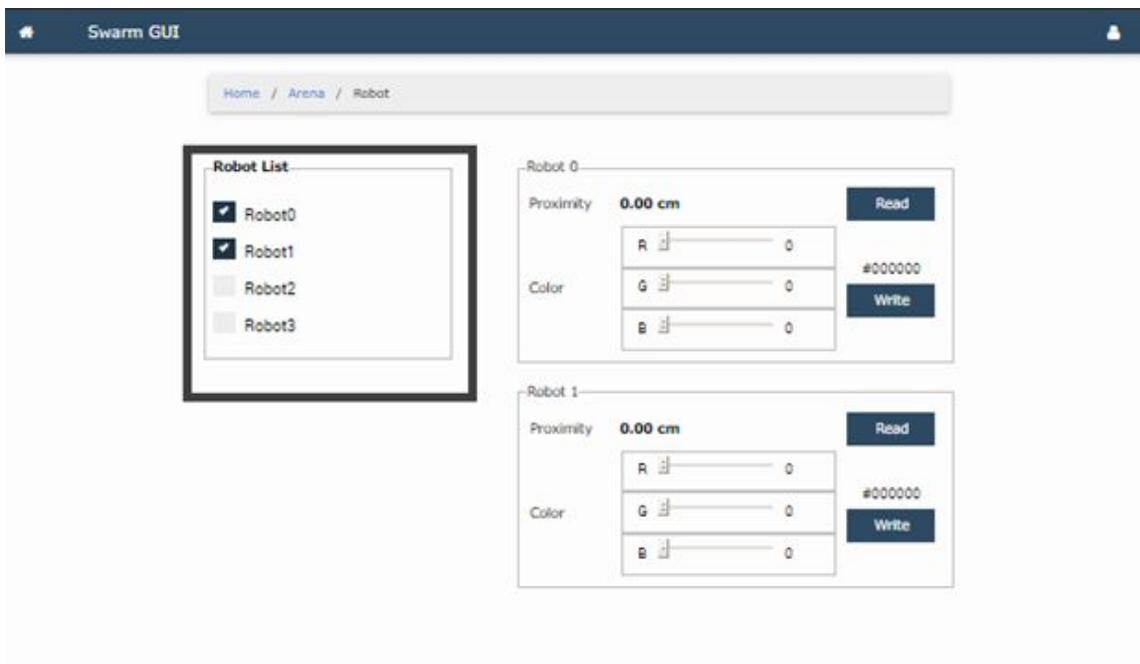
'Live video feed' panel gives a live video feedback of the movements of the robots.

3.1 Manage robots

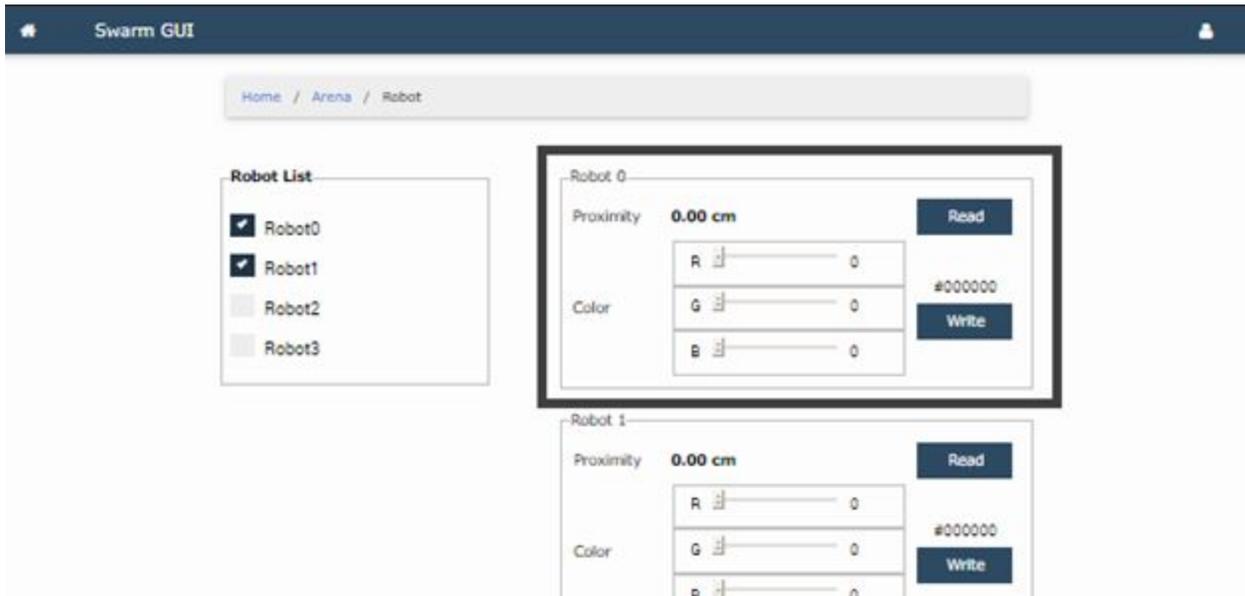
User can select 'Manage Robots' link to manage the robots.



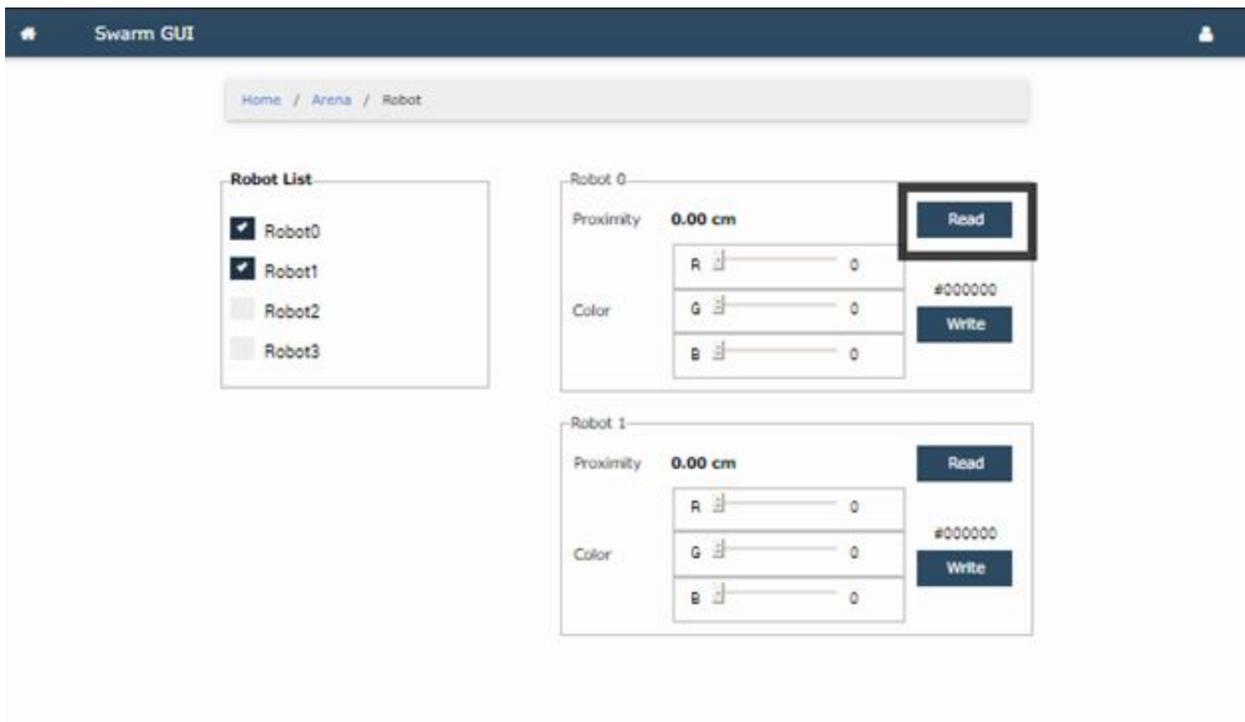
User can select robots by checking the checkboxes provided. Users can only select the active robots at that time.



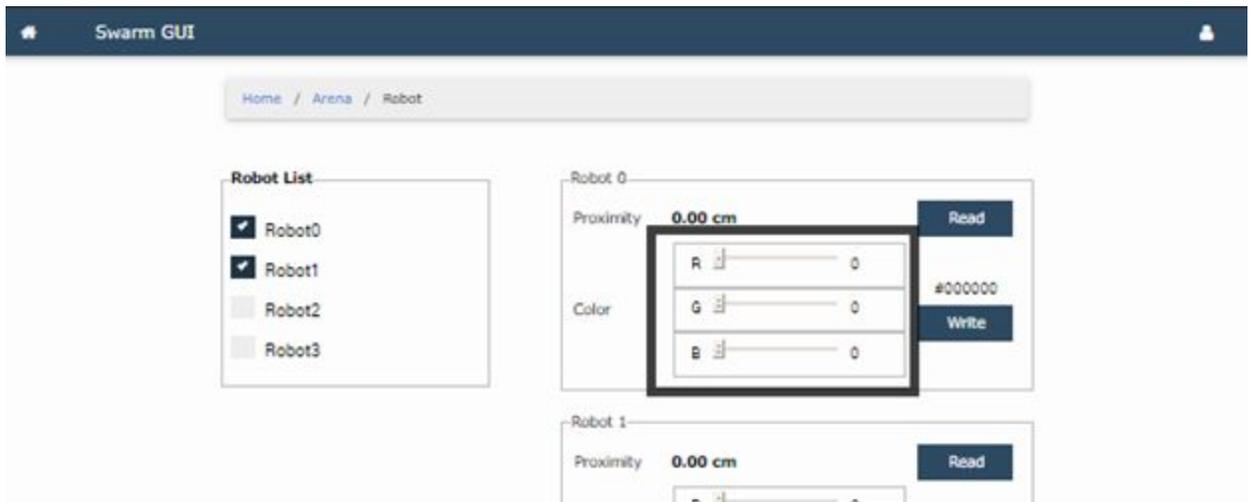
A control panel will appear for each robot when the checkbox is checked.



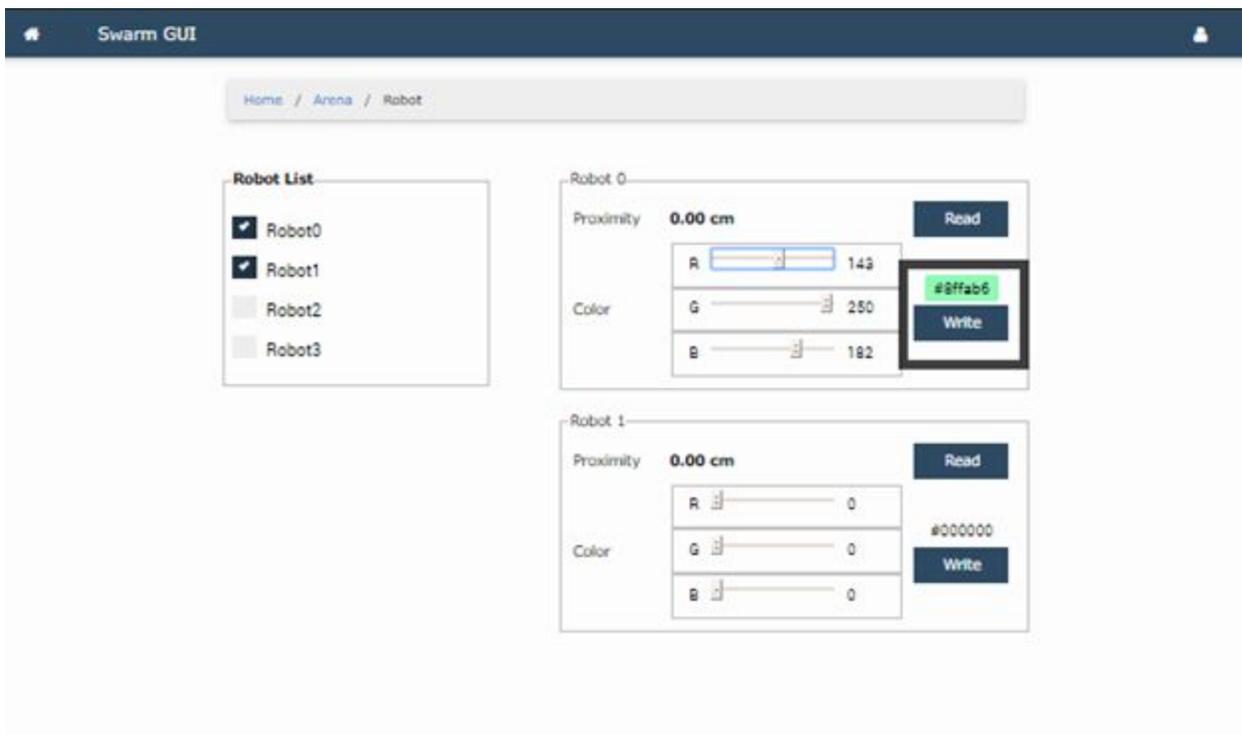
Users can read the proximity values by clicking the 'Read' button in the right of the proximity value.



User can change the color of the robot using the color panel which is highlighted in below image.



The color they selected will display in the color box and can be send to the robots by clicking the **'Write'** button.



API Interface

End users can obtain information from the robot and send instructions to the robot by sign in to the user portal and send HTTP GET requests to API endpoints defined as follows:

Root URL: <https://swarm-gui.tk/api/v1>

Note: Root URL may subject to change/update

Scan and identify available robots

Request:

/GET **{Root URL}**/robots/scan/:arenaId

Parameter	Description
arenaId	ID of the simulation arena. (Ex: 1000)

Example Response:

```
{
  "auth": {
    "arenaId": 1000,
    "upToken": "b97fc818b0005f7a9e8e3230b670a535"
  },
  "topic": "scan",
  "type": "reply",
  "jobId": 100001,
  "data": {
    "ips": {
      "0": "192.168.43.102"
    }
  }
}
```

Obtaining the proximity reading of a robot

Request:

/GET {**Root URL**}/proximity/:arenaId/:robotId

Parameter	Description
arenaId	ID of the simulation arena. (Ex: 1000)
robotId	ID of the robot, given by the scan/robot request. This is an increment number assigned to the robots connected into simulation arena, starting from 0

Example Response:

```
{
  "auth": {
    "arenaId": 1000,
    "upToken": "b97fc818b0005f7a9e8e3230b670a535"
  },
  "topic": "proximity",
  "type": "reply",
  "jobId": 100002,
  "response": "Success",
  "data": {
    "raw": "25",
    "filtered": "25.000000",
    "robotId": "0"
  }
}
```

Send the moving path to a robot

Request:

/GET **{Root URL}**/path/:arenaId/:robotId/:pathString

Parameter	Description
arenaId	ID of the simulation arena. (Ex: 1000)
robotId	ID of the robot, given by the scan/robot request. This is an increment number assigned to the robots connected into simulation arena, starting from 0
pathString	A string that contains the path to move. Robot will execute this string char by char. F: Go forward one cell distance R: Turn right L: Turn left W: Wait for a time

Example Response:

```
{
  "auth": {
    "arenaId": 1000,
    "upToken": "b97fc818b0005f7a9e8e3230b670a535"
  },
  "topic": "path",
  "type": "reply",
  "jobId": 100005,
  "response": "Success",
  "data": {
    "robotId": "0"
  }
}
```

Update the color of a robot

Request:

/GET **{Root URL}**/color/:arenaId/:robotId/:redValue/:greenValue/:blueValue/

Parameter	Description
arenaId	ID of the simulation arena. (Ex: 1000)
robotId	ID of the robot, given by the scan/robot request. This is an increment number assigned to the robots connected into simulation arena, starting from 0
redValue	The intensity of the color red. Value should be an 8 bit integer. <i>Value: [0 - 255]</i>
blueValue	The intensity of the color blue. Value should be an 8 bit integer. <i>Value: [0 - 255]</i>
greenValue	The intensity of the color green. Value should be an 8 bit integer. <i>Value: [0 - 255]</i>

Example Response:

```
{
  "auth": {
    "arenaId": 1000,
    "upToken": "b97fc818b0005f7a9e8e3230b670a535"
  },
  "topic": "color",
  "type": "reply",
  "jobId": 100003,
  "response": "Success",
  "data": {
    "robotId": "0",
    "red": "10",
    "green": "0",
    "blue": "0"
  }
}
```

