Software design description – Hardware

System overview:

\*A general overview of how the whole system should work.

System architecture:

\*A summary of how the arduino and raspberry pi are setup and communicating with each other. Should get a gist of how the hardware system is setup.

Detailed system design:

**MBot**

**Variables:**

**Functions:**

void \_delay(float seconds):

Input:

Output: None

Description:

void move(moveDirection direction, int speed):

Input:

Output: None

Description:

void moveForward():

Input:

Output: None

Description:

void moveBackwards():

Input:

Output: None

Description:

void turnLeft():

Input:

Output: None

Description:

void turnRight():

Input:

Output: None

Description:

void stopMotors():

Input:

Output: None

Description:

void collision():

Input:

Output: None

Description:

void autoTurn():

Input:

Output: None

Description:

void isr\_process\_motorLeft(void):

Input:

Output: None

Description:

void isr\_process\_motorRight(void):

Input:

Output: None

Description:

int checkSensors():

Input:

Output:

Description:

String getOrientation():

Input:

Output:

Description:

int autonomousDriving(int currentState):

Input:

Output:

Description:

void bluetoothDriving(char nextState):

Input:

Output: None

Description:

void setup():

Input:

Output: None

Description:

void \_loop():

Input:

Output: None

Description:

void loop():

Input:

Output: None

Description:

**Raspberry pi**

**Variables:**

**Functions:**

sendPositionRequest(x, y, sessionID, state, collisionFlag):

sendImageRequest(x,y):

bluetoothInit():

class CalculatePosition:

\_\_init\_\_(self):

terminate(self):

run(self, speed, newDirection):

class ReceiveBluetooth:

\_\_init\_\_(self, client):

terminate(self):

run(self):