**Time Line : Park-IT**

|  |  |  |
| --- | --- | --- |
| **TASKS** | **SUB TASKS** | **DEADLINE** |
| Design Time (Client) | * Google map   + GUI: The background of the home page will be a map   + Implement: Using google map javascript api * Menu Items   + Search textbox     - GUI: textbox that could accept info from user     - Implement: text would be sent to back-end node.js files   + Search button     - GUI: button for textbox     - implement: trigger the event for sending data to backend files   + Current location     - GUI: Will show the current location at beginning     - implement: google map js api * Navigation options   + Function: Navigation from current location to user’s destination   + implement: google map js api * Parking lot map(inside)   + Structure of the inner parking lot   + Empty spot will be shown to user   + Once user click an empty spot, navigation function is available |  |
| Run Time (client) | * Regions : Logical grouping of map , such as MSU, which also contains lat long details of MSU   + Map regions     - Function : returns the region objects based on the radius limit   + Park regions     - Function : returns the park region objects based on the region selected * Markers   + Function : Pin/Marks the parking zones based on the park regions. * Navigation   + Function: show the path from current location to the target location   + Function : show the path from current/selected location to the selected/nearest park region. * Inner parking lot   + Function: Navigation from current location inside the parking lot to the target empty spot   + Sending info to server side so the empty spot the user is going would not be visible to other users |  |
| Server | * JSON objects for storing Regions and Parking information   + MAP JSON     - Database/Server : data containing the MAP region details   + PARK JSON     - Database/Server : data containing the PARK region details |  |