No.	**A. Android Remote Controller Module** Functional Specifications	MDP Supervisor Signature / Date
C.1	**The Android application (AA) is able to transmit and receive text strings over the Bluetooth serial communication link.** Note: You can use the AMD tool to help verify that your AA has successfully achieved bi-directional data transfer.	**Completed**
C.2	**Functional graphical user interface (GUI) that is able to initiate the scanning, selection and connection with a Bluetooth device.** E.g. when the Connect button is touched, a list of available devices is presented to the user for selection. Once a device is selected, a connection is established with the device. You can use C.1 to show evidence of a successful connection.	**Completed**
C.3	**Functional GUI that provides interactive control of the robot movement via the Bluetooth link** (e.g. move forward, left and right). The interactive control of the robot movement can be done using several labeled buttons (minimal requirement), appropriate touch gestures, button cum device tilt or any other method you can think of. You can use the AMD tool to demonstrate control of the virtual robot movement. Caution: Manually entering different string commands in a text box to control the robot movement is not a valid implementation of this requirement.	
C.4	**Functional GUI that indicates the current status of the robot** (e.g. stop, moving, etc). You can implement this using a TextView box (minimal requirement). You can use the AMD tool to simulate information update by devising your own string-based protocol representing the various possible status of your robot. Note: Your TextView box must only display selective information and not all the text data that is being streamed to Android tablet.	**Completed**
C.5 (new)	**Functional GUI to enter Waypoint & Robot Start Coordinates** Touch-the-Map based interaction to update of Fastest-Path Waypoint and Robot Start Coordinates. This interaction allows users to select the fastest path waypoint and robot start coordinates by touching the required grid location on the virtual map GUI (i.e. the map in C.6). Note that entering these coordinates by typing in the values is not acceptable. In addition: 1. You need to show that the selected X and Y coordinate values can be transmitted out via the Bluetooth serial link and received by the AMD tool (This can be verified via the receive text display on the AMD). 2. You need to show your waypoint or virtual robot's graphical position in C.6 is updated once the coordinates are entered. Note: The virtual robot in the AMD tool cannot be re-located by these coordinates. You can only control the robot on your own map.	**Yet to Completed**
C.6	**2D display of the maze environment and the robot's location.** E.g. you can create a drawing canvas on your GUI where a 2D grid map (minimal requirement) showing all the positions of the known obstacles, the current location of the robot and its heading. The display can be update either manually or automatically as in C.7.	**Completed**
C.7	**Functional GUI that provides the selection of Manual or Auto updating of graphical display of the maze environment.** E.g. you can use a toggling labeled button (minimal requirement) to achieve this	

	toggling selection. *Manual Update* means the Grid map refreshes at the interactive request of the user by pressing an update button (minimal requirement) or touching the grid map. *Auto Update* means the Grid map is updated (as a background thread) automatically based on a regular interval or whenever there is a change in state over at the robotic module. Note: The AMD tool can be used to verify that you have correctly implemented both the Manual and Auto mode. Sending the string "GRID" to the AMD tool will get it to send you the current Grid Map data that can be used to render your Grid Map. The "Send Grid" button on the AMD tool can be used to simulate an auto update when the robotic module senses a change in the grid map state.	**Yet to Completed**
C.8	**Functional GUI that provides two buttons that supports persistent user reconfigurable string commands to the robot.** These buttons when pressed send a predefined string via the Bluetooth link. The predefined string can be changed by the user interactively. Once the new string has been entered, it will remain persistent or non-volatile. That is, if your Android app is closed and then re-started, the reconfigured string will remain the new string and not revert back to some default value.	**Yet to Completed**
C.9	**Robust connectivity with Bluetooth device.** Your Android application (AA) must not hang up if connectivity with the Bluetooth device is temporarily lost (e.g. by executing a Disconnect at the AMD tool after connection has been established). Your AA should automatically re-established connection automatically once the Bluetooth device connects with the AA again (e.g. by executing a Connect again at the AMD tool after connection was earlier broken with a Disconnect).	**Completed**
C.10	**Extension beyond the basics.** Novel robot movement control - Demonstrate novel remote motion control of the robot using other sensing capabilities of the Android tablet (e.g. Tilt sensing or continuous touch control) AND/OR Go beyond your basic 2D grid map display. For example, you could provide a button-activated toggle between your basic 2D grid map display and a 3D first person-view or a 2.5D view of the current known map and the robot. See visual samples in the Android briefing slide 10.	**You mad bro?**