clear descriptions of all components in their system design

all aspects of the planning, design and implementation of system components system architecture

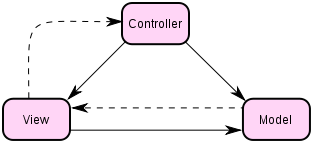
create clear, readable descriptions of the technical accomplishments

**Android Remote Controller Module**

**Application Architecture: Model View Controller (MVC) Pattern**

**The *Model***represents our application’s state. Remember, objects have state (properties) and behaviours (methods). And so applications have state (the model). In essence, models are just big name-value objects that dispatch events when its state changes. 

***Views*** are the things you see and interact with. They bind to the model by registering themselves as observers. When the model changes, the view is notified and it updates itself.  
  
When a user interacts with the view, it sends events to**a *Controller***. The controller is responsible for handling the input logic. It interprets the user gestures, updates the model, and may send messages back to the view. Below is a typical MVC diagram showing this relationship.

[](http://www.therealjoshua.com/2011/11/android-architecture-part-3-the-model-in-mvc/mvc_diagram/)

Essentially MVC pattern is:

* Model: What to display
* View: How it’s displayed
* Controller: Formatting the model for display and handling events like user input

We explored using MVC in Android, use of the State Pattern, Data Access Objects for persisting data, and Commands for sending and receiving data using a Bluetooth service.

Android UI creation using layouts, resources, activities and intents is an implementation of the MVC pattern.

**Application Components**

* Bluetooth Component
* Scan, Display, connect
* Bluetooth Robustness
  + Auto request permission to access Bluetooth
  + Auto connect when application starts
  + Remember devices that connected previously, and automatically connect it back
* Accelerometer Component
* Able to control the robot movement using button and motion control with certain angle
* Map Grid Component
* Display current 2D maze environment states, auto mode and manual mode
* Display list of robot states
* Able to calculate and showcase the robot movement and positon it is facing
* Settings Component (Shared Preference)
* Functional two button store persistent string

References:

Android Dev API Guide –Bluetooth  
<http://developer.android.com/guide/topics/connectivity/bluetooth.html>