

# ECSE 211 Group Meeting 1: Project Management

## Purpose:

To succeed in any project, all members of the team must work together to a common agreed upon goal. Communication is key! We will examine tasks to be done along with tips and tricks.

## Tasks to be done:

- Audit the provided documents
  - Project description
    - Note things that are unclear
    - Highlight any part of the document that you feel needs to be clarified
  - Capabilities doc
  - Constraints doc
  - Requirements doc
  - System doc
- Using the capabilities doc, determine everyone's skills
- Assign roles
  - Project manager
  - Hardware / Testing team
  - Software team
  - Documentation
- Write a list of next steps

## Tools and tricks to help you on the way:

### Communication and organizing:

- Slack, the team messaging service: <https://slack.com/>
  - Provides messaging services between multiple people, supports file sharing, also you to keep a detailed log
- Google Calendar: Useful for synchronizing everyone, has a Slack plugin
- Doodle, useful for scheduling when people are available



- Dropbox
  - **A must** use as Professors will be using this to access your documents
  - Allows multiple users to share documents
  - Your documents still need a revisioning system
  - DO NOT USE FOR CODE!

## Managing tasks

- Gantt Project: Provides gantt chart support
  - **A must** use as Professors will expect you to present the gantt chart
  - Tutorial for Gantt Project: <https://youtu.be/5rHCSa5ad34>
  - General info for Gantt chart <http://www.gantt.com/creating-gantt-charts.htm>

## Programming collaboratively and noting issues:

- Git & Github: <https://github.com/>
  - **A must!**
  - We delivered a lecture on how to use git and github.
  - Provides a facility to document things and mark issues
- Follow a style guide : <https://google.github.io/styleguide/javaguide.html>
  - This ensures everyone is programming the same way
  - Helps keep things clean and readable
  - Everyone should be able to understand the code
- Java docs: <http://help.eclipse.org/neon/index.jsp>
  - Can be done directly in Eclipse
  - Helps document your code
  - Generates useful documentation that will be part of your deliverables
- UML Documentation:
  - Provides a way to visualize your architecture
  - Generating uml directly from source code  
<https://marketplace.eclipse.org/content/eclipse-uml-generators-incubation>
  - Drawing UML and other block diagrams, integrates with drive, github:  
<https://www.draw.io/>
- JUnit: <http://junit.org/junit4/>
  - Useful for testing code without putting it on the robot
  - Can be helpful in testing math, filters, behaviour of classes, etc

