Team Names	Raspberries
Members	• Abdullatif AlAbduljaleel • Aaron Barge • Liam Kolber • Cooper Timmerman • Ryan Close • Wenle Feng
Description	Our product will be based on a soil humidity testing unit that can collect data on soil moisture content. This data will then be parsed and displayed via a web interfaces to the user. The goal is for our farmers to better understand their soil. After collecting enough data they will be able to better understand their crops and what is needed for a full harvest.
Vision Statement	To bring smart technology to agriculture.
Motivation	In Boulder and around the world farming plays a key roles in economy and people's lives as sources for food and energy. Water management is key in any farming operation, canales, ditches and tonnels have be built all over the world to bring water to farmers. Onces that water has been brought to the farm it should not be washed, by monitoring the soil moisture we hope to find the optimal times of day to water plants, as well as how much water is optimal for plant growth.
Risks	• Hardware issues • Lack of experience • Software bugs.

Risk management Plan	• Hardware issues
	  ○ Start early and fail fast in order to learn quickly.
	  ○ Look up other similar projects.
	• Lack of experience
	  ○ As a CU Boulder student we have free access to Lynda.com and can take courses on what we may be lacking.
	• Software bugs.
	  ○ Keep a clean github page and version control, roll hardware out in steps to keep new variables limited.

Version Control	• Github
	• Numbered and letter rev control for file management.
	  ○ Engineering reves will be numbers.
	  ▪ Rev 1, Rev 2, Rev 3…
	  ○ Lettred rev for production.
	  ▪ Rev A, Rev B, Rev C...

Development Method	Agile.
Collaboration tools	Google calendar, docs, github, GitKraken, text message.
Proposed Architecture	• Raspbian operating system Minibian, or Raspbian • Python, and the Python virtual environment • I Flask, a Python-based web micro-framework • uWSGI as the application server for Flask • Nginx light-weight web server • Possibly use Skeleton to make the web UI look better • Use RPi GPIOs pins as digital input and outputs to connect to our sensors. • Use a DHT11 temperature sensor, humidity sensor may change. • SQLite database to store sensor data • Cron job to store sensor data. • Use the Google Chart API to create visual representations of the sensor data • Javascript/JQuery to add interactivity to web pages • Plotly for graphical analysis of sensor data