# Project Plan

## 2.1 Project Management/Tracking Procedures

Which of agile, waterfall or waterfall+agile project management style are you adopting? Justify it with respect to the project goals.

**We’ll be using AGILE as our project management style for this project. Tasks will be posted to 2 week incremented sprints & assigned to group members as the group sees fit. This will make breaking down tasks easy for the group. Having a way to modularly assign tasks will be crucial for any development the group does, as well as research/physical tasks needed for the project. AGILE will also help keep the group on track with its 2 week sprints.**

What will your group use to track progress throughout the course of this and the next semester. This could include Git, Github, Trello, Slack or any other tools helpful in project management.

**The group will use a few different softwares including: Gitlab, Monday.com, & Google Drive. Gitlab will be used for our codebase, allowing any dev work to be shared amongst the group. Monday.com is our AGILE tool, equipped with a kanban board, sprint planner, & other useful tools. Finally, google drive will be used to share any design documents, canvas assignments, & other documentation the group compiles.**

## 2.2 Task Decomposition (Bryanna)

In order to solve the problem at hand, it helps to decompose it into multiple tasks and subtasks and to understand interdependence among tasks. This step might be useful even if you adopt agile methodology. If you are agile, you can also provide a linear progression of completed requirements aligned with your sprints for the entire project.

**Task 1**

* **Determine hardware requirements**
	+ **Research different smart greenhouse designs**
	+ **Decide on design for our test bed**
	+ **How many and types of plants we want to test**
	+ **Decide on sensor placement based on what components we are monitoring**

 **Task 2**

* **Determine software requirements**
	+ **Database**
		- **What do we want to monitor**
		- **How often do we want the database to pull data**
	+ **App**
		- **What do we want the app to show**
		- **Decide how the app shows it**
		- **Decide on different ways the user can access (i.e. app, website)**

**Task 3**

* **Cost analysis**
	+ **Fixed costs: Lights, sensors, greenhouse structure, soil**
	+ **Variable costs: Plants, power costs(i.e. electricity, water)**

 **Task 4**

* **Create proposal for Iowa State University**
	+ **Determine optimal design**
	+ **Create final design documents**

## 2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria (Devon)

* completion of a rev 1 for hardware design, allowing us to start developing software for this hardware.
* software implementation to get us to the point we have a user interface and can start adding functionality to other hardware and software components such as sensors and long distance wireless communication.
* individual sensor hardware and software will each be its own milestone.
* reaching a MVP (minimum viable product).
* creation of a working database.
* creation of a working phone app to have a user friendly interface.

## 2.4 Project Timeline/Schedule (Julia)

• A realistic, well-planned schedule is an essential component of every well-planned project

• Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity

• A detailed schedule is needed as a part of the plan:

– Start with a Gantt chart showing the tasks (that you developed in 2.2) and associated subtasks versus the proposed project calendar. The Gantt chart shall be referenced and summarized in the text.

– Annotate the Gantt chart with when each project deliverable will be delivered

• Project schedule/Gantt chart can be adapted to Agile or Waterfall development model. For agile, a sprint schedule with specific technical milestones/requirements/targets will work.



## 2.5 Risks And Risk Management/Mitigation (Jasen)

Consider for each task what risks exist (certain performance target may not be met; certain tool may not work as expected) and assign an educated guess of probability for that risk. For any risk factor with a probability exceeding 0.5, develop a risk mitigation plan. Can you eliminate that task and add another task or set of tasks that might cost more? Can you buy something off-the-shelf from the market to achieve that functionality? Can you try an alternative tool, technology, algorithm, or board?

**Task 1 - potential risks while determining hardware requirements:**

* **Researching different smart greenhouse designs:**
	+ **Risk factor: 0 -** There is no risk in looking at other designs for inspiration.
* **Settling on a testbed design:**
	+ **Risk factor: 0.5 -** The initial testbed design may end up changing quite a bit as testing progresses.
	+ **Risk mitigation:** Although there is nothing we can do to mitigate risks prior to testing, we will at least be able to correct any design flaws when we do begin testing.
* **Choosing plant types:**
	+ **Risk factor: 0.2 -** If any problem arises with a certain type of plant, we are still able to choose another.
* **Sensor layout:**
	+ **Risk factor: 0.2 -** Like most of the project, sensor layout is subject to change as the project progresses. There is little risk other than any problems that may go unforeseen later into the design process.

**Task 2 - potential risks while determining software requirements:**

* **Database design:**
	+ **Risk factor: 0.2 -** The database design can be changed at any point in the process, so there is no major risk.
* **App design:**
	+ **Risk factor: 0.2 -** As with the database design, the app design can also be changed at any time.

**Task 3 - potential risks during cost analysis:**

* **Fixed costs:**
	+ **Risk factor: 0.4 -** Because we will compile our list of fixed costs before ordering everything, we should be able to make sure we stay under our budget. A potential risk is finding out later that we need to order a new part, however this should not be too much of an issue.
* **Variable costs:**
	+ **Risk factor: 0.5 -** The variable costs will be the hardest to control, as the more we are able to optimize our design, the less power we will consume.
	+ **Risk mitigation:** We need to spend a lot of time working down the variable costs in the late stages of the design process.

**Task 4 - potential risks while creating a proposal:**

* **Determining optimal design:**
	+ **Risk factor: 0.4 -** As designing our project will be an iterative process, we may make mistakes in determining the optimal design before enough testing has been done.
* **Finalizing design documents:**
	+ **Risk factor: 0.1 -** As long as we follow good design practices, there should be no risk.

## 2.6 Personnel Effort Requirements (Jake)

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort in total number of person-hours required to perform the task.

I wasn’t really sure if I was supposed to summarize Bryanna’s tasks or what we needed to do for the entire project so I made 2 tables

| Task | Summary | Person-hours |
| --- | --- | --- |
| Determine Hardware Requirements | Researching/designing greenhouse testbed for project including sensors and produce | ~30 hours |
| Determine Software Requirements | Researching software requirements to utilize hardware in greenhouse | ~30 hours |
| Cost Analysis | Estimating costs for hardware/software requirements | ~3 hours |
| Create Proposal for ISU | Brainstorming and designing proposal for design | ~10 hours |

| Task | Summary | Person-hours |
| --- | --- | --- |
| Creating Test bed | Crafting physical greenhouse testbed to grow and observe progress of plants | ~6 hours |
| Installing Testbed Sensors | Place sensors around testbed to observe variables | ~3 hours |
| Integrating sensors into microcontroller | Programming microcontroller to read sensor values | ~8 hours |
| Programming Microcontroller | Programming microcontroller to react to sensor values i.e. triggering heaters/fans and water pumps | ~15 hours |
| Installing environment gadgets | Place pumps and fans/heaters around testbed to enable resource manipulation | ~8 hours |

## 2.7 Other Resource Requirements (Sarah)

Identify the other resources aside from financial (such as parts and materials) required to complete the project.

* **IKEA furniture -> that can be turned into a greenhouse/grow box**
	+ **Depending on the greenhouse/furniture chosen to be our green house we will need glass**
* **optional growing information for the plants**
* **Camera**
* **Sun/grow lamp**
* **Timer for the lamp**
* **Camera**
* **Plants**
* **Moisture monitoring equipment**
* **Water hose**
* **Water tank holder**
* **Thermometer**
* **Soil**
* **Plant food**
* **Identify a platform and language for the application**