Team 1 - VIR - Atlas

Visible Infrared Atlas

Project Proposal & Plan

1. Introduction (20 points)

1.1 Project Overview and Statement of Proposal

Our team proposes to build an accurate visible and NIR (near-Infrared light) spectrum (or Color-Infrared) mapping software specifically for STELLA (Science and Technology Education for Land/Life Assessment), that will cartographically include and/or display STELLA's other sensor readings in a user friendly and visually appealing GUI (Graphical User Interface).

1.2 Project Scope and Objectives

Our team has been presented with the unique opportunity to work with a newly developed and nearly patented piece of hardware called STELLA, a handheld device developed by NASA researchers mimicking Landsat technology. Its purpose is to easily survey and record numerous environmental conditions for a given geographic area (relatively small) with an accuracy comparable to Landsat. STELLA's various sensors measure conditions such as humidity, air pressure, and surface temperature. However, our team is most interested in its visual and near-infrared light spectrum readings.

The data that STELLA generates presents similar applications and opportunities that the Landsat program provides. This data is fundamental in many fields like ecological analysis, preventative forest fire management, search and rescue surveys, land-use and development planning, and so much more. The best way to utilize and understand the data for almost any of these applications is to visualize it into a map, a feature STELLA isn't currently capable of.

2. Risk Management Strategy (50 points)

2.1 Risk Table

#	Risks	Category	Probability	Impact	RMMM
1	Data file size underestimation	PS	10%	1	2.3.1.1 2.3.2.1 2.3.3.1
2	App interface is not user-friendly	BU	30%	2	2.3.1.2 2.3.2.2 2.3.3.2
3	Task deadlines will not be met	BU	30%	3	2.3.1.3 2.3.2.3 2.3.3.3
4	Lack of sufficient software documentation	CU	50%	1	2.3.1.4 2.3.2.4

					2.3.3.4
5	End product will differ from original plan	PR	70%	2	2.3.1.5 2.3.2.5 2.3.3.5
6	Miscommunication leading to incorrect features	PR	50%	3	2.3.1.6 2.3.2.6 2.3.3.6
7	STELLA project discontinued	PR	10%	4	2.3.1.7 2.3.2.7 2.3.3.7
8	Technology will not meet expectations	TE	50%	3	2.3.1.8 2.3.2.8 2.3.3.8
9	Complications with transferring data to app	TE	10%	2	2.3.1.9 2.3.2.9 2.3.3.9

Category values:

PS – Product Size Risk

BU – Business Impact Risk

CU – Customer Relations Risk

PR – Process Risk

TE – Technology Risk

DE – Development Environment Risk

ST – Risk Associated with Staff Size and Experience

Impact values:

4 – catastrophic

3 – critical

2 – marginal

1 – negligible

2.2 Discussion of Risks to Be Managed

We've assessed that our most pressing risks are miscommunication between team members and our technology not meeting our current expectations. If the team isn't communicating effectively, there's a chance we'll have different understandings of the features we'd like to include. Should we have an issue with our drone or STELLA, we'd have difficulty collecting data to work with. These risks have both a high impact and a relatively high probability of occurring. We've made sure to cover the mitigation, monitoring, and management of these two risks.

Another of our highest impact risks is the discontinuation of STELLA and our deadlines not being met. Our software will be designed to create maps exclusively from the data output of a handheld STELLA device. Should the STELLA project be discontinued, we wouldn't be able to collect data to create a map, and therefore wouldn't have any reason to make mapping software for it. The probability of this is very low, but the severity is fatal. Additionally, there's a chance for miscommunication between team members leading to incorrect features. Since this would result in a different end product, it has a severe impact on our project.

Our other high probability risks include a different end product and a lack of documentation. We'll likely run into a problem that requires a change to the current plan. While this could alter some of the product's features, it has a marginal impact on our software's overall functionality. Another issue we're likely to experience is a lack of documentation for STELLA. Since we're working with nearly patented hardware, there might not be sufficient support and documentation for the tools we're using.

Our less pressing potential issues include underestimating data size, complications transferring the data, a non-intuitive GUI, and a lack of documentation. We may get such a large amount of data from STELLA that it's difficult to handle and transfer to our software. If we have other difficulties parsing data, it will be difficult to map it. Additionally, since we're using a new and nearly patented device, there's a chance we won't have enough documentation and support to use them effectively in our project.

2.3 Risk Mitigation, Monitoring, and Management Plan

This section describes what is to be done to avoid each risk (Section 2.3.1), how the team will monitor its activities to detect when a risk becomes a problem (Section 2.3.2), and what will be done for each risk if it becomes a problem (Section 2.3.3).

2.3.1 Risk Mitigation

1. **File size underestimation**
    a. We can avoid this issue by understanding what kind of data we're working with and how to handle that data efficiently.
2. **App interface is not user-friendly**
    a. Potential users will test the interface to ensure that it is easy to use and intuitive.
3. **Task deadlines will not be met.**
    a. With our weekly meetings, our goal is to set bite-size goals for each team member to reach by the next meeting so that no team member gets overwhelmed.
4. **Lack of sufficient software support**
    a. Keeping logs of any changes made to our software is key to understanding any bugs we can/will encounter. Additionally, our team members will be ready to lend a helping hand to other team members who need it.
5. **End product will differ from original plan**
    a. With a Gantt Chart and weekly meetings, our team will always have clear goals laid out for what the software will accomplish.

   b. If certain problems arise, the team will meet and discuss alternative solutions and be flexible with our plans.
6. **Miscommunication leading to incorrect features**
   a. We can avoid miscommunication by meeting weekly on Discord and keeping an open line of communication between all team members.
   b. Minutes that outline tasks delegated to each team member are made for each meeting held to discuss the project.
7. **STELLA project discontinued**
   a. Since STELLA is closely related to the LandSAT, we can use data online from the LandSAT to continue our project.
   b. We can consider alternative sets of sensors for collecting data. It would be best to avoid this due to the potential increase in cost.
8. **Technology will not meet expectations**
   a. Before we begin developing our software, we can examine the data we collect from our STELLA device and ensure it meets our mapping requirements and expectations.
9. **Complications transferring data to app**
   a. Before we begin developing our software, we can examine the data we collect from our STELLA device and ensure we're prepared to handle the format of the data.

2.3.2 Risk Monitoring

1. **File Size Underestimation**
   a. We can monitor how long it takes for our software to process a given set of data.
   b. We can monitor the file size of the data we get from STELLA.
2. **App interface is not user-friendly**
   a. We will be able to monitor whether or not the interface is user-friendly by testing it throughout development.
3. **Task deadlines will not be met**
   a. We can track our progress and compare it with our proposed schedule.
4. **Lack of sufficient software support**
   a. During development, we can monitor how well we document our software's features.
5. **End product will differ from original plan**
   a. As we progress, we can record the changes we make from the original plan.
6. **Miscommunication leading to incorrect features**
   a. We will have regular meetings to ensure we all have the same understanding of our software expectations.
7. **STELLA project discontinued**
   a. We will stay in regular contact with the developers of STELLA.
8. **Technology will not meet expectations**
   a. By comparing STELLA's data to the drone's image, we can regularly assess how accurate it is.
   b. We will run regular test flights to ensure the drone is working properly.
9. **Complications with transferring data to app**
   a. We will check the data we enter into the mapping system for errors and inconsistencies.

2.3.3 Risk Management (Contingency Plans)

1. **Data file size underestimation**

   a. If the file size becomes an issue, we may need to reconsider our approach to handling the data. Possibly dealing with chunks rather than the entire file.
   b. We can also increase the size of the storage.
2. **App interface is not user-friendly**
   a. Two contingencies are the redevelopment of the GUI and better product documentation.
3. **Task deadlines will not be met**
   a. We will need to reconsider some of our additional features so we can make the final delivery date.
   b. We may also need to reallocate resources/time towards tasks that are taking longer to develop than originally planned.
4. **Lack of sufficient software support**
   a. While working with unfamiliar software and documentation, we may want to consider alternatives that have better support.
   b. In the case of a lack of documentation, we'll schedule more time for proper software documentation.
5. **End product will differ from original plan**
   a. We will need to reassess our position/goals throughout the course of this project. The Final Report will likely include whatever alternative paths we pursued as well as the reasoning for them.
6. **Miscommunication leading to incorrect features**
   a. We will hold a meeting to realign our team goals and reassess the nature of application features.
7. **STELLA project discontinued**
   a. STELLA collects similar data as LandSat. We may be able to swap STELLA's data for data collected by LandSat.
8. **Technology will not meet expectations**
   a. See **item 7.a** for a relevant contingency plan.
9. **Complications with transferring data to app**
   a. We will need to consider alternative methods for transferring the data to the app or how it is handled. Some considerations are: working with the data in the app itself, having a server that carries out any calculations/transformations, and a front-end application that queries the server for relevant information.

3. Schedule (20 points)

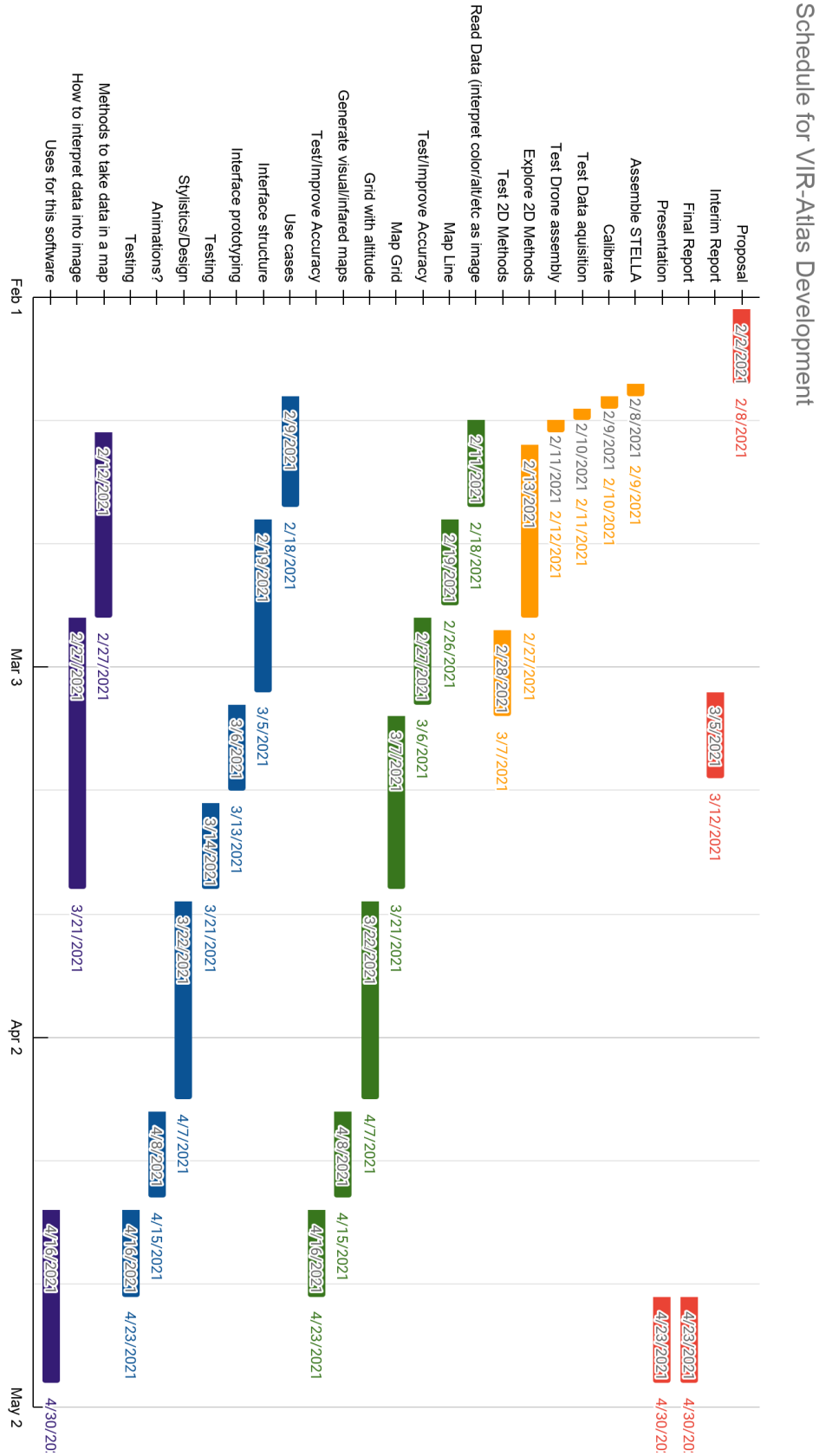
3.1 Task List

1. Paperwork (Included for timelining)
   a. Proposal
   b. Interim Report
   c. Final Report
   d. Presentation
2. STELLA Assembly
   a. Assemble STELLA
   b. Calibrate
   c. Test Data acquisition
      i. Ensure calibration is correct and data files can be obtained
   d. Test Drone assembly
      i. Ensure Drone assembly is completed
   e. Explore 2D Methods
      i. Test methods of obtaining grid point data to go with VIR data

   f. Test 2D Methods
3. Back-End Development
   a. Read Data (interpret color/alt/etc. as an image for one line of data)
      i. Be able to interpret a single line of data correctly
   b. Map Line
      i. Be able to interpret multiple lines of data correctly into a map of a straight path
   c. Test/Improve Accuracy
   d. Map Grid
      i. Interpret data points to a grid and generate a map
   e. Grid with altitude
      i. Generate a map that accounts for accuracy with altitude
   f. Generate visual/infrared maps
      i. Full map developed with proper grid positioning and color generation
   g. Test/Improve Accuracy
4. Front-End Development
   a. Use cases
      i. planning for basic interactions with software based on use
   b. Interface structure
      i. Interface planning
   c. Interface prototyping
      i. Purely functional GUI
   d. Testing
   e. Stylistics/Design
      i. Finalizing the style of the GUI
   f. Animations
      i. Any animated interactions with the GUI
   g. Testing
5. Research
   a. Methods to take data in a map.
      i. Investigate methods to place VIR data to points on a grid
   b. How to interpret data into image
      i. Investigate methods of generating the most accurate image from the method we retrieve data.
   c. Uses for this software

3.2 Timeline Chart

Gantt Chart (with dates)


Schedule for VIR-Atlas Development
Proposal
Interim Report
Final Report
Presentation
Assemble STELLA
Calibrate
Test Data aquisition
Test Drone assembly
Explore 2D Methods
Test 2D Methods
Read Data (interpret color/alt/etc as image
Map Line
Test/Improve Accuracy
Map Grid
Grid with altitude
Generate visual/infared maps
Test/Improve Accuracy
Use cases
Interface structure
Interface prototyping
Testing
Stylistics/Design
Animations?
Testing
Methods to take data in a map
How to interpret data into image
Uses for this software
Feb 1
Mar 3
Apr 2
May 2
2/2/2021 2/8/2021
2/8/2021 2/9/2021
2/9/2021 2/10/2021
2/10/2021 2/11/2021
2/11/2021 2/12/2021
2/13/2021 2/27/2021
2/28/2021 3/7/2021
2/11/2021 2/18/2021
2/19/2021 2/26/2021
2/27/2021 3/6/2021
3/7/2021 3/21/2021
3/5/2021 3/12/2021
2/9/2021 2/18/2021
2/19/2021 3/5/2021
3/6/2021 3/13/2021
3/14/2021 3/21/2021
3/22/2021 4/7/2021
4/8/2021 4/15/2021
4/16/2021 4/23/2021
2/12/2021 2/27/2021
2/27/2021 3/21/2021
4/23/2021 4/30/20
4/23/2021 4/30/20
4/16/2021 4/30/20

3.3 Resource Table

Task	People	Hardware & Software	Special
Paperwork	All	N/A	N/A
Assemble STELLA	Sophia		STELLA, Arduino, Drone, CircuitPython, Github
Back-End Development	Franklin, Marisa, Sophia, Timothy	Gitlab, Python	STELLA Data, Drone Image
Front-End Development	Tyler, Tenise	Gitlab, Python	N/A
Research	All	N/A	N/A

4. Project Resources (10 points)

4.1 People

1. Paul Mirel, a primary developer of STELLA
2. Tyler Charity
3. Marisa Loraas
4. Timothy Goetsch
5. Franklin Keithley
6. Tenise Stansfield
7. Sophia Novo-Gradac

4.2 Hardware and Software Resources

1. Discord
    a. For team collaboration and meetings
2. Gitlab
    a. Private source code management
3. Python
    a. Language for front and back end development
4. PyCharm
    a. preferred IDE for Python development

4.3 Special Resources

1. STELLA
    a. Hardware to take data
2. CircuitPython
    a. Language STELLA's software is written in
3. Arduino
    a. IDE for exporting code to STELLA
4. Drone
    a. Take birds-eye images and data
5. Github
    a. CircuitPython code for STELLA is stored here

5. Appendices (5 points)

This section contains the breakdown of individual contributions and any additional information such as meeting logs that you would like to include in the project proposal and plan.

Here is a sample of what we plan our meeting minutes to look like.