Hydro Citizens

Requirements Specification Document

Version 1

November 21, 2017

Sponsor:
Dr. Benjamin L. Ruddell

Team Mentor:
Dr. Eck Doerry

Team Members:
Luis Arroyo
Logan Brewer
Ryan Ladwig
Kelli Ruddy

Accepted as baseline requirements for the project:

For the client: ____

Signature Date

For the team: ____

Signature Date

Table of Contents

1. Introduction **3**

2. Problem Statement **4**

3. Solution Vision **5**

4. Project Requirements **6**

5. Environmental Requirements **14**

6. Potential Risks **14**

7. Project Plan **16**

8. Conclusion **17**

9. Glossary **18**

1. Introduction

Our waterways and wetlands have an important role in our everyday lives and the more knowledge we have about them, the better. Hydrological data is information about rivers, reservoirs, lakes and other waterways. This data is important when it comes to flood prediction/prevention, water management, and public education/knowledge. If we have more data, we can more fully understand how rainfall fluctuation affects a specific area. Flooding is one of the most destructive aspects of increased amounts of rainfall and can result in loss of homes and even death. Although we can never fully predict how severe a flood will be, we can better understand how increased rainfall will affect that area if we have the data. Water management has to do with measuring river flow and runoff as well as infrastructure design. Infrastructure design refers to how roads, neighborhoods and even cities are built to handle increased amounts of rainfall. Public education/knowledge is directly connected to water management in that if a community more fully understands how increased amounts of rainfall could affect their area, this may influence how they vote for public officials. They may be more likely to vote for someone who understands why this data collection and design is so important. Right now the United States Geological Survey (USGS) collects most of the hydrological data in the U.S., but the problem with this is that they only collect data from large waterways and reservoirs. This is because the gauges they set up to collect this data are extremely expensive so it is less practical to collect data on smaller waterways or ephemeral rivers, which are rivers that do not flow year-round. We need to understand how rainfall fluctuation affects certain areas not monitored by the USGS. Without this knowledge, certain areas could be negatively affected by increased amounts of rainfall. Having more data is also important because then we can more fully educate the public on hydrological information and its importance. Having the public understand why we need to continue collecting this type of data is important, especially when it comes to our solution.

Figure 1.1 *- Our gauging solution*

Dr. Benjamin Ruddell is our sponsor for this project. He has worked on a prototype application of this project that focused on a web based solution whereas our solution will focus on developing a mobile application.

Our solution to this lack of data is to develop a mobile application that allows users to take pictures of a gauge which, unlike the USGS's expensive equipment, will be a wooden post in the ground and a PVC pipe with red and white stripes as seen in Figure 1.1. The wooden post will have a QR code attached that can be used to uniquely identify the site and to provide more information to passersby.

Once the user has taken a photo of the painted PVC pipe, our image processing algorithms will add a horizontal line to where it believes the water line is in the photo. The user may adjust this line to get a more accurate water level.

We will use the user's phone's geolocation in order to plot exactly where the photo has been taken and this location will be stored with the photo upon submission. From there we will store the image, location, and water height data in our local database. We will then take the water height data and send this to our local HydroServer and then it will be joined with the main national HydroServer which is where a lot of national hydrological data is stored. Once the user has uploaded their information, we will give them feedback on their submission, which may include previous data points collected at that location or a graph showing how their contribution helps this data crisis. One key feature that separates our application from anything else will be the offline capabilities. Some of these gauges and waterways will be in areas without cell service or internet and this is where our application comes in. The user will be able to take a photo and store the information associated with the photo on our application and will be ready for submission when they are back in an area with coverage. The user should still be able to receive some confirmation that their data was recorded, and be notified when their recordings are synced to our database.

2. Problem Statement

The United States Geological Survey is the main contributor for hydrological data in the United States. They collect gauge height and streamflow from large lakes, rivers and reservoirs. This data is collected from large water gauges, seen in figure 2.1, every 15 to 60 minutes. The data is then stored onsite and sent to a USGS location for analyzing every 1 to 4 hours. The onsite data is sent via telephone, radio or satellites. This data can be available online within minutes or after a full day of data collection, the information is transformed into a daily summary and released online for the public. The daily summary can be a graph showing *stage* in terms of feet. The user can also view the summary of all available data for that specific site which may consist of more graphs. To access the daily summary takes a minimum of 6 clicks to view. Another problem with the current system is the funding to maintain these gauges. Yearly maintenance of a USGS water gauge is around $14,000 - $16,000.

Figure 2.1 *- USGS water gauge at Salmon Falls Creek located in Nevada*

Since 2013, there have been 18 discontinued water gauges due to lack of funding and there are currently 9 water gauges in danger because funding is unlikely to be continued for these gauges. Funding for these gauges comes from Federal, State, Tribal, and local agencies and funds specifically appropriated to the USGS. More efforts are going into finding new ways to get funding but until this happens, more and more gauges will be discontinued. Without these gauges, certain regions will suffer because information on their local waterways will not be collected and analyzed. The problem with the USGS's data collection method, although extremely accurate due to the equipment, is the cost and scarcity of data points. These water gauges are expensive to maintain and install and will result in fewer data points if they cannot get funded and must be shut down. If lack of funding continues, our hydrological data collection will begin to deplete. Also, USGS only collects data at larger waterways due to the cost of water gauges. Not enough data is being collected from ephemeral rivers and smaller waterways so we do not have enough knowledge on how these waterways react to situations of increased rainfall. The more data we have, the better we can prepare for floods.

The daily summary given to users is not very effective in that it takes at least 6 clicks to get to a graph and summary of a specific gauge. The graph is also not descriptive and often-times difficult to understand for a person not familiar with hydrological terms. This makes it difficult for the community to have an interest in the collection of hydrological data. We hope through our solution to provide more data points on smaller waterways as well as on waterways where USGS water gauges were previously located and to get the general public more involved in the collection process.

3. Solution Vision

Now that we have described the problems that we aim to solve, we are going to explain how we aim to solve these issues. We are going to build a mobile app that will allow users to collect hydrological data from all of the small waterways that the USGS is unable to collect data from. We will streamline the current process and allow the user to quickly and efficiently collect this data. Some specific features that our solution will include are:

- An easy-to-use mobile application that has a user-friendly interface
- A simplified way for users to access all hydrological data, including data they have collected.
- Promotion of an inexpensive way for collecting and sharing hydrological data.

We will be implementing this in a multitude of ways. The information that we are going to be collecting is hydrological data, which will include:

- The location where the observation was made.
- The date and time of the observation.
- The height of the water from the observation.

This data will be used to educate the public on how rainfall affects runoff in smaller waterways. It will also be used to fill missing data points for smaller waterways and ephemeral streams where data isn’t currently being collected. Our local database will be storing the images, water depth, location, and date for all the data entries. From there, we will take the hydrological information and send it to our local HydroServer. Finally, all of the data from our local HydroServer will be federated to the national HydroServer.

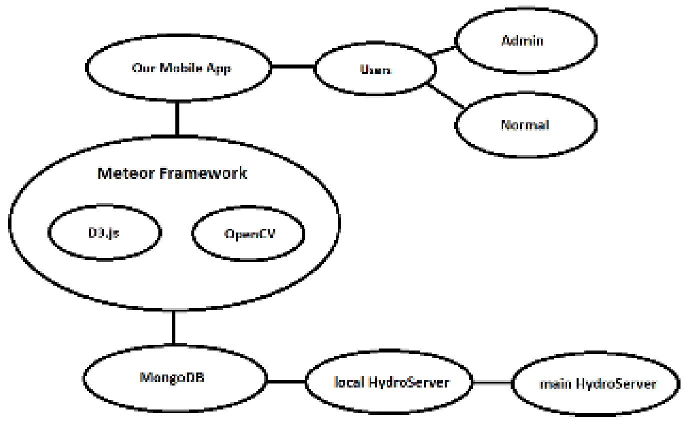

Figure 3.1 - System Diagram

This will be a great improvement on the client's current web-based version. The mobile application will be used to spread knowledge of this project out to many more people, as well as to speed up the general data collection process of the client's current solution.

This will allow us to collect more hydrologic information data in a faster period of time than it would have before we implemented our solution.

When looking at all of the possible solutions, we had several mobile application frameworks, databases, computer vision and data visualization tools that we could choose from. After much thought and research, we have decided to begin initially with Meteor, D3.js, OpenCV, and MongoDB, however these are subject to change. Our solution will enable users to collect this very important hydrologic data very quickly and efficiently around the entire world.

4. Project Requirements

4.1 User Accounts

When creating a web based solution or a mobile application, user accounts are an important feature to have. A user account is important so that a user can have a personalized experience. An admin account is also an important type of account to have. We want to make sure the mobile application provides users a few features that the current web solution does not.

4.1.1 Functional Requirements

Functional requirements of user accounts include the ability to:

- Submit hydrologic data such as photos and notes on the photo's to a database.
- Perform quality control on their photo submission.
- Change notification settings within in the application.
- Allow a user to view their past submissions.
- Search for gauges in their area.

The user should be able to submit a photo with notes to a local database. The photo may be taken directly on the camera in the application or the user should be able to go into their camera gallery and select a photo for submission. Once the photo has been taken or selected, the user should be able to add notes about specifics regarding the photo before submission. This may include the weather or other things going on at the time that the picture is being taken.

Once the photo has been taken and the notes have been added, the user should be able to go through a quality control check. This will include adjusting a line placed on the photo by our image processing algorithms to determine where the water level is. After the user has decided the line placement is where it should be, they may submit it to the local database.

The user will receive push notifications daily, weekly or monthly based on their preference. This can be changed within the applications settings. The reason behind this is to retain interest in our application and make sure the notifications are not becoming a nuisance to the user.

The user must also be able to view their past submissions if they have submitted to the application before. This allows them to recall submissions at areas they have previously recorded data for.

If the user is not familiar with the area, they should be able to search for a gauging station in that area. This will happen using the user's geolocation and the stored locations of the known gauges.

4.1.2 Performance Requirements

The user should be able to learn how to use the application easily. We want to limit how many clicks are necessary to get to a desired location. This will help in retaining the user's interest. 3 or fewer clicks should allow the user to get anywhere on our application. A user must be able to retrieve their past submissions in .5 seconds. This process should be almost instantaneous.

4.2 Database

The database for our application must be able to store user submitted photos, hydrological data and notes. Our database must be able to connect with a larger database to be able to federate to the national HydroServer. When deciding on the requirements for our database we kept in mind the three main things listed above that our database needs to do.

4.2.1 Functional Requirements

Some functional requirements of the database include the ability to:

- Store photos, notes, and hydrological data.
- Store the location of each gauge.

Our database must be able to store photos, notes and hydrological data that are all produced by the user. It must store this information because we want to be able to show the user their past submissions as well as to be able to submit the user's information to the HydroServer so that it can be analyzed. Our local database will store each of these requirements. We also want to be able to store each gauge's location in our local database. We will see what gauging station the user is at using their geolocation and then store the site of the station in our database. These are the main parts of what information needs to be stored in our database.

4.2.2 Performance Requirements

Looking at our functional requirements helped us to decide that our performance requirements for our database must include:

- Scalability
- Connectivity
- Reliability

Our database must be able to scale easily in the case of an increased amount of users and user submissions. It must handle this well. The connectivity of the database must be fast so a user is not waiting on it in order to view past submissions. The database must retrieve a user's submitted data in .5 seconds. The database must be reliable in that it must be able to give users their past submissions without the fear of loss of data.

4.3 Data Visualization

Data visualization is a feature to display data through graphs and charts. This feature is important because the hydrological data, that the user collects, needs to be presented to them in a way that is visually pleasing and easier to digest than just showing them plain text. When looking at how data visualization will be incorporated into our project, we split it into the functional and the performance requirements.

4.3.1 Functional Requirements

When analyzing the functional requirements, the two major requirements are:

- Connecting to a database while online in order to display the requested information.
- Storing data on the user's phone in order to display graphs while being online and offline.

First we will look at the online component. One thing that comes along with this is being able to connect to the National Weather Service (NWS). The NWS is used to provide weather forecasts, weather warnings, and to educate the public. We will use the NWS to show the user rainfall data and see how the rainfall data is connected to the amount of water in the small rivers and ephemeral streams. The NWS is an online feature because it is constantly updating new information. It would be difficult to store the relevant information for the user without adding too much in the user's cache and overwhelming their mobile storage.

Another component that we need to handle is connecting to the National Water Model (NWM). The NWM is used to predict what water levels we be across the United States. The NWM should make a direct comparison between the data the user collects and the predictions that the National Water Model makes. This is important because it can show the user how inaccurate the NWM is for smaller waterways where the data isn't being collected currently. This would require an online connection because it would also be space consuming to store all the relevant information on the user's phone.

The next component we need to handle is connecting to the HydroServer. This will be important because this is where all the data the users will collect will transfer to. Not only does the user need to be able to see all the data they submitted, but also data from all users that are submitting data to project. This will likely be displayed in a time versus water depth bar graph or line graph.

The final component we need to handle is the offline connection. What we will do is select very specific data in order to not overwhelm a user's storage. We will take all of the data that is being stored at locations the user contributes to and store that data on the user's phone cache. With this data being stored locally, we will be able to pull from it and create a hydrograph that will show the user all the data that has been collected from that specific station. We need to be careful to store enough data that is relevant to the user without overwhelming their phone's storage.

4.3.2 Performance Requirements

When looking at the performance requirements there are two major requirements:

- Connecting to online databases to make comparisons between user data and data predictions.
- Caching specific data on user's phones in order to provide functionality while offline.

First we will look at the online capabilities. The first component is that we need to ensure that the user is able to receive a hydrograph showing a comparison between the data collected and what the predicted water level was. This needs to be done as soon as the user is done adjusting the water level and is ready to submit the image.
The second component we will perform is caching data on the user's phone so we can provide visualizations during times that they do not have a network connection. This will involve taking data from stations near the user and storing it on their phone. We will need to make sure this data doesn't overwhelm the user's storage in the process so we need to be careful about which data we are choosing. But caching is necessary. Otherwise we will not be able to provide the required hydrographs when the user is offline.

4.4 Geolocation

Geolocation is the process of identifying and finding the geographical location of an object by using geographical coordinates and measurements. It will be important to implement the geolocation into the mobile app because we need to mark the gauges that are located on the map. Through analyzing how geolocation will be incorporated into our project, we have developed descriptions of the functional and performance requirements of geolocation, as it pertains to our project.

4.4.1 Functional Requirements

Our functional requirements can be categorized into the following three major sections:

- The application must push notifications to the user.
- The user should be able to locate gauges while using the application.
- The application should attempt to identify each site based on its location.

First we will analyze the components of when to send the user notifications. One component is to send notifications when the user is within a certain distance (in meters) between the user and the next available gauge. The purpose of this requirement is to be able to motivate the user to walk a couple more meters to a small stream flow and take a picture of the water gauge. The next section is pinpoint where the water gauges are located in the map. The way this can be done is by allowing the user to find our data collection sites and pinpointing them at each gauge. The database will have the location of the streams and be able to place the markers on the map of where they are located. The purpose of this feature is to help the user to be able to find where the water gauge is and making it easier for the user to locate it. However, if the user is offline or the geolocation fails to read an accurate location, the user should take a picture of the QR Code scan. Once they have an online connection, the QR Code should be able to match to the location's site.

4.4.2 Performance Requirements

Our performance requirements can be categorized into the following two major sections:

- Attempt to identify each site location.
- Identify the user's location by taking a picture of the QR Code.

The application should attempt to identify each site using the location as determined by the user's phone, however if it cannot identify a particular site, the user must take a picture, with the application, of the QR code posted at the site description that will be used to identify the location of the site.

4.5 Computer Vision

A key functionality of our client's project is the ability to determine the gauge height of body of water by taking a picture of a striped pole that has been planted into the bed of this body of water. This task will require what is known as a Computer Vision Framework (CV Framework), which will run the algorithms needed to analyze an image taken by the user and predict where the water line on the pole is, a task that falls into the category of image processing. After the algorithms have made a guess as to where the water line on the pole is, the user should be able to adjust the guess to improve the accuracy of the reading. The team must also develop a more appropriate color scheme for the PVC poles, as the current red and white color scheme performs poorly in snowy conditions.

4.5.1 Functional Requirements

The functional requirements that will need to be fulfilled by our CV framework include:

- User must be able to adjust the algorithms' guess of the water height.
- The algorithms must run on an Android phone.
- The algorithms must run natively on a mobile device.
- The algorithms should not require internet access.
- Use existing algorithms developed by Dr. Ruddell's team.
- A new color scheme for the PVC poles needs to be developed.

The application will allow the user to adjust the algorithms' guess of where the water line is, which could be in the form of a line that the user can adjust with their finger. Because the user may not have access to a stable internet connection, the user must be able to record observations offline which will later be synced to the project database. Dr. Ruddell's team has already designed a set of algorithms that the team will use in the mobile application, and team will, at a minimum, adapt these algorithms so that they can be run within our mobile application.

4.5.2 Performance Requirements

The main performance requirements for the image processing algorithms can be broken down into three sections:

- New high-contrast color scheme
- Improved results in suboptimal lighting conditions
- Maintain original run times

While the team will develop a new high-contrast color scheme for the PVC poles, due to the unpredictable nature of the weather, we cannot make any guarantee that the new color scheme will be tested in snowy conditions. The new color scheme will, however, be tested, at the very least, under normal weather conditions. While the team aims to improve the performance of the algorithms under suboptimal lighting conditions, we will provide, at a minimum, a color scheme that works under the same lighting conditions as the existing red/white color scheme.

4.6 Gamification

Gamification is used in order to convert an ordinary task into something that is more similar to a game. The purpose of the gamification is to motivate users to continue performing the task when they otherwise may have stopped using it. This is important because we need to investigate the best way to retain users and keep collecting data. When looking at incorporating gamification into our application, we split it into the functional and the performance requirements.

4.6.1 Functional Requirements

When looking at the functional requirements, two major features are:

- User accounts are able to earn points and achievements.
- Push notifications are used for multiple purposes.

The first component we will look at is the user accounts. We are going to be introducing two systems that will help enhance the user experience. First is allowing user accounts to level up. This will entail giving certain amount of points to accounts for submitting data. We will also be creating specific contribution boards to each station. Through this, users will be able to get in touch with other people who are contributing to that location.

This point system will also allow to create a bounty system. This will entail giving a certain point value to stations as a baseline. From there, in certain weather conditions or if data hasn't been collected at a specific station in a long time, we will be able to increase the point value to incentivize users to go and collect data at that location.

The second component that will be added is pushing notifications. These notifications will have a setting where they are able to toggle on or off depending on if they want to keep receiving the notifications. The first notification we will add is to let the user know when they are within range of a station. This notification will be set to be on by default for every station. Once the user has encountered the station for the first time, this notification will be able to be set off because the user will already have seen that specific station.

Another type of notification that we will be sending to the user to go out and collect data when certain weather events have occurred. The first event is when it has rained. This is important because we will be able to collect data from these waterways directly after it has rained and see the effect that the immediate runoff has on the water distribution. Another weather situation is notifying users when it snows. This is important because we can track the temperature along with the snow height across a long period of days, and see how the snow level rises and shrinks in relation to the amount of flowing water in the connected waterways.

The last type of notification we will be sending is one that asks the user to collect a data point if they haven't collected any data in the past month. This one is important to keep the user engaged and coming back to the app, even if they have forgotten about it or turned off all the other notifications. This notification will not be able to be turned off in order to ensure that the user keeps getting some form of notification to keep helping data collection.

4.6.2 Performance Requirements

When analyzing the performance requirements, it again breaks into two major sections:

- Implementing points and achievements for user accounts.
- Creating visualizations to educate the user.

First we will look at points. The points will be in direct relationship with the pictures the user submits. We will be keeping a standard value for all the stations but when it is raining or when a station hasn't collected data in a while we will be increasing this value, creating a bounty on stations essentially.

Secondly, we will implement data visualizations. These will need to be generated instantly in order to show the user the effect that their data has, as well as to educate them on how runoff or rainfall directly affects the amount of water in a specific location. This specific visualization needs to work online or offline. However, when the user is online, we will have access to a much larger amount of data, including data from the National Weather Service and the National Water Model. We will be able to display information from these websites upon the user's request when they have a network connection.

4.7 Maintainability

In the context of our project, maintainability is the act of designing software such that long-term maintenance of the software is kept at a minimum. While we have no control over the external services that our application and database rely on, we will document all of our code and we will deliver a separate document to the client that describes how each of the parts of our project fit together. Specifically, we will deliver a document to that client that will:

- Outline each of the external services that our application relies on.
- Describe the role of each of these services within our project.
- Describe the expected/required inputs and outputs of each service.
- Describe and outline how all of the parts of our project communicate with each other.

The document will fully describe how each of the parts of our project fit together, so that if changes to one of the services that our application uses cause our application to lose functionality, a new team should be able to look up the role that that service plays in our project so that they can determine an appropriate solution.

5. Environmental Requirements

The most important environmental requirement is that we are developing a mobile application. We will be developing this application for Android users, but we aim to develop our application in a way that should allow for a relatively easy port to iOS, made after the original Android application is developed. Our application will also need to communicate, and be compatible with, the other technologies that are a part of Dr. Ruddell's system, namely:

- The HydroServer network
- Database that stores pictures and user-collected water data
- Web Server that manages user accounts

The application should communicate with the local database to store hydrological data collected with the application and to also store the pictures from which the data was collected. The database should then be proofread and formatted before being submitted to the Hydro Server. The application should also be able to connect to a web server that will allow users to keep track of their accounts and log into the mobile application.

6. Potential Risks

In this section of the document, we will describe the potential risks that could prevent us from implementing the minimum requirements for this project. Through discussions with the client and researching the parts of our project and how they will fit together, we have developed a list of the potential risks to the development of our project:

- The National Water Information System data format is outdated.
- Compatibility with the HydroServer.
- Changes to other applications and services that our mobile application will rely on.
- Geolocation accuracy.

6.1 National Weather Information System Outdated

USGS provides access to their collected water data through the National Water Information System (NWIS). We will be using this data to show the user how their observations compare to those of USGS. They have acknowledged that their data distribution services, that use tab-delimited data files (RDB), are severely outdated. They have announced plans to update the format in which their data is available, but have not made an official announcement as to how the data access will be modernized or when these changes will take place. Possible data file types include Keyhole Markup Language (KML) or Excel spreadsheets. While they have not announced when these changes will take place, their website claims that data in the existing RDB format will be provided for an extended period of time, even after the new data formats are made available, and for this reason we consider these potential changes to be of medium risk to our project.

6.2 Connecting to the HydroServer

HydroServer is a software that is used for publishing hydrological data. Our local database will need to connect to the HydroServer and send the hydrological data that the user recorded from the gauge. However, the local database could be incompatible with the HydroServer. Because of this, we will need to ensure that our database will be able to connect to the HydroServer and submit data in the proper format.

In order to mitigate this risk, we plan on setting up our own HydroServer for development purposes. We will test the communication between our local database and our test HydroServer before connecting to a live HydroServer.

6.3 Changes on Application Services

The application will be using APIs such as the National Weather Service (NWS), National Water Model (NWM), and other packages that the mobile application will be using. However, any of the APIs or services that we will be using could be modified by the original owner. If a package or packages changes, the mobile application will need to adapt to the change.

In order to mitigate this risk, we will have well-documented and well-structured code. We want to make it easier for other developers that will continue maintaining and working on this project. With having well-structured code, developers will have less of an issue to modify the application services.

6.4 The Accuracy of the Geolocation

One of the key requirements for this project is to use geolocation in order to locate where the gauges are on the map. We want the user to be able to find these gauges easily and in a reasonable amount of time. However, there could be issues when trying to pinpoint where those gauges are for the user. Since we will likely be using a package for our geolocation needs, we are relying that the package will provide a longitude and latitude with reasonable accuracy. According to some research, there have been accurate readings from within 3 meters (~10 feet) of a target location, but there have been other reports that this reading is not always accurate. The reason it obtains the incorrect information is because it depends on the signal that is transmitting from the GPS satellites to the phone, and the signal between the phone and the GPS satellites is not always reliable.

This is a low risk because we can use the QR code to identify each site's location. The QR Code can store data such as longitude and latitude. When the user takes a picture of the QR Code, it will obtain that data and it will be stored in the database. The QR Code will provide more of an accurate reading compared to the geolocation because we will know where the picture was taken and if the picture they took was at a streamflow.

7. Project Plan

In this section of the document, we will describe our plan to implement the minor key requirements into the application. Figure 7.1 illustrates our plan to implement the milestones for our project. The milestones that we have are:

- Parts Feasibility
- Back-End Development
- Front-End Development
- User Testing
- Code Refactor
- Presentation

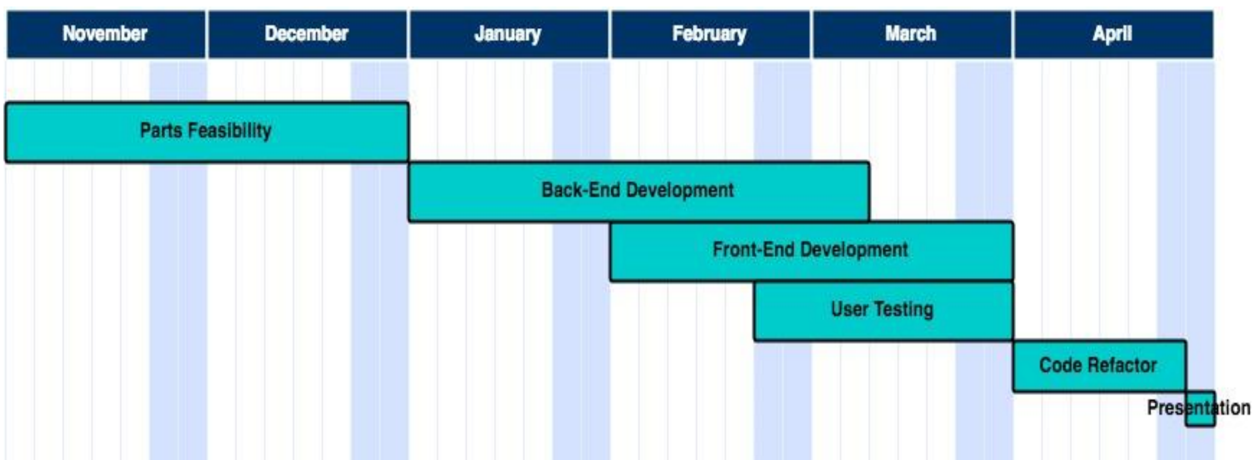

Figure 7.1 – Project plan Gantt chart

At the current stage of the project, we are working on the parts feasibility. Each member of the team will be implementing a different functionality of the application separately so that they will become familiar with their assigned functionalities. The functionalities are: connecting to the database, implementing basic data visualization, geolocation, and image processing. Once each functionality has been implemented separately, we will combine them into one coherent application.

From the start of January until the mid-March, everyone will focus on working on the back-end development of the key functionalities. The back-end development will be responsible for the interaction of the database and the performance of the application. With the database, we need to be able to add and store data to the local database and then submit this data to the HydroServer.

From the start of February until the end of March, we will be working in front-end development while still implementing the back-end development. The objective of the front-end is to create friendly user interface design that is easy to use. The reason that this will involve some of the back-end development is because when the user presses a button, it will need to have some functionality from the back-end such as being able to submit hydrological data.

During the month of February until the end of March, we will perform a series of user tests. The users will be students and professors that would be interested in the application and will focus on selecting users who might have a vested interest in using our application. We will ask users what they like and don’t like about the application, and if it easy to use. With the feedback we receive from the client and the users, we will implement these features while taking feasibility into consideration. It is for this reason that we will need some time to implement both the front and back-end of our application.

From the start of April until the presentation date, we should have all the functionalities working. Here we will start refactoring code and making sure that it is structured in a way that is relatively easy to maintain for future developers. We will then be prepared to present our mobile beta at the symposium.

8. Conclusion

Hydrological data collection is important because without it, we are risking the safety of homes, properties and lives. As the USGS loses funding for their stream gauges, we must come up with a way to continue the collection of the large waterways that USGS collects data on as well as smaller waterways they do not collect data on. This is where our project comes in. With our solution we will be able to collect more data on smaller waterways and provide the people who live in these areas with better warnings in the case of floods and other hydrological disasters. Our application will allow users to be able to take a photo of a “stream gauge” which will be a PVC pipe with colored lines on it. Once the photo is taken the user will be able to set a line where the water level is and submit that to a database to be analyzed. This requirements document will help our team to decide what the most important features of our application will be. We have now outlined the key requirements and will strive to complete each and every one by the end of the year. As a team, we are confident that we will be able to fulfill all of the requirements listed throughout this document and end the year with a fully functioning application that will be able to provide more accurate data points that could help save lives.

9. Glossary

Application Programming Interface (API) – A set of tools and protocols for building software applications.

HydroServer – A software application used to publish hydrological data.

Computer Vision Framework – Set of software and applications that perform image processing.

National Weather Services – An agency of the United States federal government that provide weather forecasts, warning of hazardous weather, and other weather-related features.

Stage – Depth of water at a particular point.

Water gauge – Physical site that is used to collect water data.