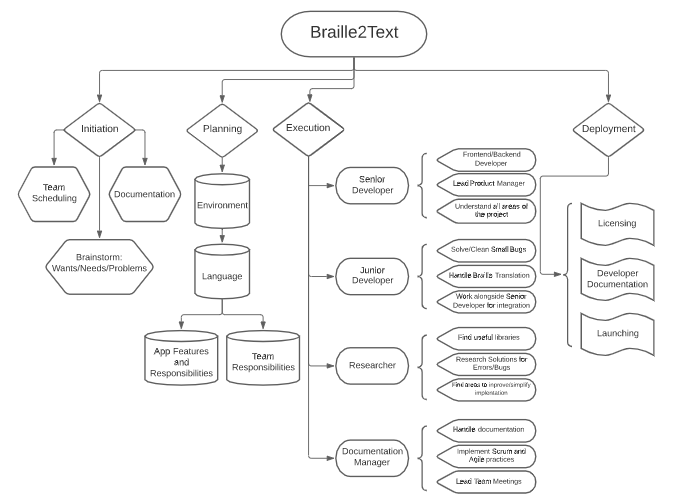
1. ***Project Charter***

For my project, I am creating a Braille Translation Application. This will be made for the latest version of Android. This Application will be able to scan written or printed braille and translate it into the English language and grammar. The goal is to do this in a user friendly mobile app and translate in near real-time.

1. ***Plan Your Project***

******

1. ***Monitor Progress***

We will use the SMART test to set the specific reporting for the project. Specific: An area for improvement could be communication and accountability. We need to be sure to communicate constantly and be sure the entire team is on the same page. Accountability can be improved as some members have slacked and were unable to meet requirements/benchmarks. This allowed other members to assume the responsibility, causing more issues within the team.

Measurable: We should be sure every couple of weeks we make consistent progress. This can be done by setting benchmarks to meet. When we reach the scheduled amount of time, we need to have an essential team meeting understanding the next steps to take.

Assignable: We must give every member a niche or task to be responsible for. This must be done with integrity and accountability. This will ensure each member is helping and the workload is spread amongst the group.

The Senior Developer is tasked with knowing all there is to know about the process of the project. They must know all parts to ensure each language and process works seamlessly together. The Junior Developer works alongside the Senior Developer to assist and make sure implementation is agreed upon. The Junior Developer will branch off and work on a more specific area in the code. The Researcher is tasked with finding the best intuitive way to implement the project. This includes searching languages to use and the correct libraries to integrate. The Documentation Manager will handle all documentation regarding the application process. They will be sure the team is communicating and are using correct Scrum/Agile practices.

Realistic: During each meeting we must relearn and specify the goals for the given benchmark. Keeping realistic goals are important to progress. Life can include all sorts of complications and readjusting to those are important to continue consistent progress.

Time Related: We want each sprint to be 2-3 weeks long and have around 5 sprints. Each sprint will have a main benchmark to meet and if not met, it will bleed into the other sprint’s timeline. This also allows some sprints to be short giving more time to other sprints. Each sprint is for each phase in the work breakdown structure.

1. ***Phases***

* **Initiation**
  + In this stage, we must make sure to let the team know the schedule. This will be to ensure constant communication within the group during the time of the project. Meetings will be at least weekly.
  + Initial documentation will be needed as we need to understand where to begin our project. We must also start on the requirements needed and details on the features we want in the final product.
  + We will need to brainstorm what we want the product to do and what features we want to implement. This will include the main problem we want to solve and use of our product.
* **Planning**
  + We will first need to know what environment we will need to develop in. Since we are developing an Android application, we will use the native environment or Android Studio.
  + Next our team will decide on a language to use or what languages would be most intuitive to implement. We will use Python for the translation as it is one of the most powerful languages and has multiple useful libraries for photo manipulation and scanning. We will use another number of languages for frontend development as well.
  + Lastly, we must create separate roles for each team member. This is to evenly separate the workload and to ensure each member is working on an area they are best suited for. Giving a niche to each member also allows for holding each person accountable for the work everyone does.
* **Execution**
  + This stage we will finalize the responsibilities for each member. The Senior Developer is tasked with knowing all there is to know about the process of the project. They must know all parts to ensure each language and process works seamlessly together. Tasked with assisting the Junior Developer with any errors with the code.
  + The Junior Developer works alongside the Senior Developer to assist and make sure implementation is agreed upon. The Junior Developer will branch off and work on a more specific area in the code. In this case, the actual translation of the data. They will also be tasked to clean and simplify the errors in the final product.
  + The Researcher is tasked with finding the best intuitive way to implement the project. This includes searching languages to use and the correct libraries to integrate. They should continue working to improve and simplify the implantation. Helping with solving computation errors.
  + The Documentation Manager will handle all documentation regarding the application process. They will lead meetings and ensure the steps needed to proceed and make progress. They will be sure the team is communicating and are using correct Scrum/Agile practices.
* **Monitor & Control**
  + This will be done by mainly the Senior Developer and the Documentation Manager. It is to ensure all benchmarks are met at scheduled times and track performance of the team.
* **Closing**
  + This step will lead to deployment of the application. Licensing will need to be done and multiple bug fixes will need to be done as well. This is also the best time to simplify the code and improve performance.

1. ***Retro***

* The main reason IT Projects fail for our project, communication. Communication is key to any project. We struggled with keeping communication between all team members. Some members would go days without contacting the rest of the group.
* An issue was interaction for us. During the times of COVID, we must meet only online. We did not meet in person and this caused some interesting problems. This included getting the program and environment on all members separately and explaining how to implement things. Tech can be hard to explain and understand when not together.
* Accountability was also an issue as some members would not finish the responsibilities of their position. This caused other members to assume responsibility of other members. This made it difficult to meet benchmarks and finish on schedule.
* A problem with understanding and staying on the same page within the entire team. Some team members may want to implement a function differently or dislike the path of the project. We mitigate this by making sure constant communication is met.
* Lack of skill and unwillingness to learn was another issue. When some team members are confused or do not know how to do something, some will give up and will not take the necessary steps to learn. Team members can help but some people will not learn or have motive. This causes other members to assume the responsibilities.