

# 1. Talent Inventory

Each member of the group should introduce themselves, and describe their experience, interests, and relevant expertise. This will help to adjust the project idea and technical choices to the group profile. Include in your report a brief sentence to summarize the experience, interests, and technological expertise of each member.

**Jesse:**

C#, C, Python, Unity, Java, Javascript, HTML, CSS, Kotlin (Learning), Scratch.

**Dorian:**

C, Python, Javascript, HTML, CSS

**Travian:**

C, Python, HTML/CSS/Java script, Japanese

**James:**

C, Python, Assembly

**Daniel:**

C, Python, HTML/CSS/JAVA

**Cole:**

C, Python, HTML/CSS/JAVA, C, Assembly

## 2. Roles

As described in the “Team Project Instructions” document, there are four roles (manager, quality assurance, reporter, and presenter), and everyone should rotate through these roles at least once. Each member of the group should have no more than one role per week unless your team has less than four active members. On top of these roles, every member is expected to work on coding and writing the project deliverables. After everyone has rotated through the roles at least once, the group can decide to keep rotating or to assign permanent roles to specific team members.

Your report should describe your role rotation schedule until everyone has rotated through the roles at least once.

*For now, we will rotate once every week for simplicity, moving “down” in the order shown below. So next week, James will be Presenter and Jesse will be Quality assurance.*

**James:**

Quality assurance

**Jesse:**

Reporter

**Cole:**

Manager

**Daniel:**

Presenter

**Travian:**

Presenter

## 3. Meeting Schedule

You should have in-person weekly meetings starting in the 2nd week of classes until reading week (see the Course Schedule on Canvas). No meetings are required during Spring Break (if applicable). Most project deliverables are due on Sunday, thus, meeting near the end of the week can be an opportunity to work

together to wrap up the documents and plan for the following week. You can also schedule another meeting at the beginning of the week for planning purposes and additional quick meetings just for follow-up if the group feels the need.

Document your meeting schedule in the team report.

Thursdays 6:45 pm, Cline Library room 249, be there or be square.

## **4. Communication Plan**

Discuss in your group the use of communication tools like Slack (<https://slack.com/>) or Discord to communicate about project-related issues during the week.

Also discuss how you are going to deal with disagreements on design choices. E.g.: "We will make decisions by unanimous agreement" sounds nice, but a "2/3 majority" may be more helpful in moving things along.

Also discuss how to deal with non-performing team members. Coming to basic agreements about some of the most contentious issues of group work before these issues have surfaced provides a basis for discussing and dealing with them calmly. Putting these issues on the table as legitimate for group discussion also makes more explicit the co-responsibility that all group members have toward one another and helps to avoid some difficult situations.

Finally, remember that the actual contribution of each member toward each deliverable of the project should be reported and that peer evaluation has a significant impact in your grading. Please, don't let your team down!

We have a group chat using text for communication, and we will deal with disagreements using a majority vote, if we ever tie the votes, we will use a coin flip for fairness.

## **5. Project Idea**

Now it is time to discuss project ideas and scope. After deliberating in the group, use the weekly report to describe what your product is going to be and the main proposed functionalities.

**Some remarks about the project scope: I would like to remind you that the focus of our course is on software engineering techniques, so the project does not need to be technologically sophisticated (e.g., dealing with large volumes of data, adopting cutting-edge technology, etc.). You don't want to spend most of your time trying to figure out how to accomplish the implementation itself or learning a new framework or programming language. Just manage the risks. The more important aspect of the project is that it must be valuable for someone. The most common choices for this course are mobile apps and web systems. I don't want you to finish the course and say "hey, this is a proof of concept that shows that this technology works," I prefer "hey, this is a prototype of a product that has value for someone and that is well-designed for future evolution." Of course, it is very hard to come up with an idea for a helpful product out of the blue. Don't worry, you can change your mind or pivot over the semester. We will have classes to discuss how to determine if your idea is worth pursuing.**

**Also begin thinking about what technologies will be used. You can change later on, but it is worth deciding about the initial infrastructure. Think about programming and scripting languages, database technologies, frameworks, libraries, protocols, GUI, running platforms, development environment, remote services, etc.**

#### **Technologies:**

This will be a web-design based project, using HTML, Javascript, CSS.

#### **Ideas:**

Calendar/ Health Tracker, with a focus on fitness oriented people.

Music creating and sharing applications, with a focus on people who are interested in not only creating music but sharing music.

## **6. Next Steps**

**Now it is time to plan the next steps (who/what/when). We'll be discussing tools to manage your work in the coming weeks but start with a plan that describes who will work on what in the next week**

We will be communicating and collaborating on decision making using a group chat. Ideas for the project are currently being discussed and decided on. When we have decided our general idea, we will work on determining a good target audience and implement from there.