**Team Planning**

Once you select your team, identify the strengths and weaknesses of your team. First consider the interests of your team members and write your names in the respective columns based on your preferences. After you complete this table, discuss everyone’s preferences and converge on assignments that make everyone happy.

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Top preference | Not my top preference, but happy to help | I’d prefer for someone else to take the lead on this role |
| Project Management Lead |  |  |  |
| Requirements Lead |  |  |  |
| Design Lead |  |  |  |
| Implementation Lead for front end |  |  |  |
| Implementation Lead for back end |  |  |  |
| Configuration Management Lead |  |  |  |
| Testing Lead |  |  |  |
| Documentation Lead |  |  |  |
| Demo and presentation Lead |  |  |  |
| System Administrator Lead |  |  |  |
| Other: |  |  |  |

In addition to identifying the lead roles for team members, identify the languages and technologies that you are considering with pros/cons of each with respect to the project. For instance, C is probably not a good choice for a project to create an online store. On the other hand, perhaps using the Android SDK is a good option for the specific project and no one has experience with it, but everyone is excited to learn more.

|  |  |  |
| --- | --- | --- |
| Language/Technology | Pros | Cons |
| Python | Easy to Implement, large community and Libraries and platform Independent | Slower Execution and High Memory Usage, Interpreter usage |
| Streamlit | No need for third-party languages like HTML, CSS, and JavaScript | Less flexibility (limited Customization) in the web design |
| MySQL | Utilized easily for Relational data | Hard to Deploy or connect with the cloud server |
| MongoDB | Flexible Schema, Best for No SQL Data | High Memory usage, Data Redundancy |
| SQL Connector | Platform Independent, Implicit data type conversion | Performance limitation compared to its alternatives (PyMySql, mysqlclients), slower compared to C libraries |
| Streamlit Cloud | Free Hosting, Easy Deployment | Limited Resources, limited customization, and No built-in database so an external database should be connected. |

Once you’ve listed all of your ideas, distribute the top choices for team members to explore before your next meeting. Ideally, every team member should explore every option, but if time is prohibitive, assign at least two team members to explore the top three options. Each team member should explore books, videos, tutorials, and a small sample project to create a summary of the pros/cons for the respective approach.

|  |  |
| --- | --- |
| Language/Technology | Assigned to: |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

After each team member explores the options, you should meet and make an executive decision on the path forward. Once you decide, you should plan to meet to work on tutorials together so that everyone is immersed into the languages and technologies that you will use on your project. You may pursue these tutorials individually to some degree, but you should also meet as a team to help each other to quickly gain experience and confidence together.

**Create a weekly meeting schedule**

A one semester project provides approximately 16 weeks for a team to create a project, but keep in mind that this includes everything from forming teams in the first week to giving demos and presentations in the last week or two. Your professor will probably provide guidance on the schedule with due dates for different deliverables throughout the semester. In this section, we focus on the weekly meeting schedule for students since regardless of the variations that occur in different sections of courses with different professors, you’ll still want to maintain a standard weekly meeting schedule to help your team members to succeed.

Exchange contact information with each other, depending on your preferred way to maintain contact. Please don’t feel obligated to share every option below, but share the ones that team members are comfortable sharing. Some possibilities are listed below.

|  |  |  |
| --- | --- | --- |
| Team member | Email | Phone |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

It’s also helpful to chat online. List the platform(s) that you will use and an ID for each member to connect.

|  |  |  |
| --- | --- | --- |
| Team member | ID for platform: | ID for platform: |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Tools to connect**

Technology is constantly changing. There are many great tools to for software engineers. Explore using the following tools. Each team member should set up an account and connect as a team.

* Github – This tool is used by millions or developers for configuration management. Visit their website, create accounts, and give it a try here: <http://github.com>
* Trello – This tool is a great way for teams to assign and track tasks. Visit their website, create accounts and give it a try here: <https://trello.com>

Teams should meet at least twice each week outside of class, but preferable more often. Teams should strive to make each team member as productive as possible. This includes sharing both positive contributions and challenges. Use the following template to review each other’s progress at least once each week.

Name:

Tasks that I accomplished this past week:

Tasks that I plan to accomplish this week:

Issues that may interfere with my success this week:

Take the time to discuss each team member’s progress, goals, and issues that may interfere with their success. Remember that each team member has a finite amount of time to work on the group project. You want to optimize the productivity of each member’s time that they put into the project. For instance, if a team member is stuck on a task that takes several hours for them to figure out and another team member could have helped to solve their problem in just a few minutes, that is an overall loss to your team. Everyone on the team should strive to make each person as productive as possible. Consider the following scenarios for discussion.

1. Team member A has never worked with a database before. They are struggling to figure out how to connect to the database, but are embarrassed to admit this to team members. Instead, they spend 10+ hours trying to figure it out to eventually learn that they needed to use a VPN when trying to connect to the database when they work from home. What could the team have done to prefer this team member from wasting so much time?
2. Team member B is not confident at coding. They wanted to take the lead on documentation in order to avoid coding. However, the instructor was very clear that everyone needs to contribute to the code and that this will be monitored by the configuration management tool and the code checked in. How should the team address this issue that Team member B wants to try to get by without coding?
3. Team member C has a busy schedule and continuously misses internal team deadlines. They work full time and take courses, so they suggest that other team members do more work. What should the team do?
4. Team member D is regularly late or misses group meetings. What should the team do?
5. Team member E does not contribute anything to the project for an entire week. There are only 15 weeks in the semester, so this is equivalent to 7% of the semester. Some team members realize that a lot may be accomplished in a week and are unhappy about their unprofessionalism. What should the team do?
6. Team member F is the only team member who is not familiar with the programming language and environment that the team chose. Since they have less experience, they think that they should do less work. This is not fair to the rest of the team. What should the team do?