Risk	Likelihood	Severity	Mitigation
GitHub repository is shut down temporarily or permanently – all of our project's files and documents are stored in a GitHub repository. If the service is shut down or inaccessible for any reason, we will not be able to access our files.	Extremely unlikely	Manageable	All of our code will be duplicated on each of our team members' personal computers, so code will not exist only on GitHub servers.
Team member gets sick or is unable to complete responsibilities – since we are dividing our work into individual modules, if one member fails to deliver complete work for any reason, we may fall behind schedule or not complete the project by the deadline	Unlikely	Critical	All team members will have a working knowledge of, or at the very least familiarity with, all other modules so that any member can work on any module if necessary.
Personal computer troubles (hard drive crashes, lost, etc) – each member will be writing his code on a personal computer, and a hard drive crash or lost computer could result in lost code.	Possible	Manageable	All code is regularly committed to our central GitHub repository, so code loss should be minimized. The university provides computer labs that can be used for code development if a member's personal computer is unusable.
Messy code – as our code grows and we add more features, it is possible that it will become hard to read and modify.	Possible	Manageable	We will review our code regularly to find potential messy areas as soon as they become a problem, so they can be fixed before they become a bigger problem.
Requirements change – some details of the requirements could change midway through the process, requiring code rewriting.	Unlikely	Critical	We will attempt to keep code flexible and reusable, so that if there is a relatively large requirements shift, we can reuse as much code as possible instead of starting over.
Modules do not communicate with each other well – as we start integrating our modules together, it is possible that they do not pass information back and forth as well as anticipated, breaking the program.	Possible	Manageable	We will begin our implementation phase as soon as possible, to find these communication errors early and fix them before the deadline approaches.