**Definition of tasks involved in the project**

This section will discuss the project scope, its break down, and whom is responsible for delivering each section. The goal of this software is to provide the user a simple and fun racing game. To do this, we want the user to be able to create an account to save all of their records, select and customize cars from a provided list,

The project’s work flow will progress as follows:

* Build a Login Screen – This screen allows the user to enter a username, password, and try and login. If the user has not created an account, then a link for the user to register. If the user is able has an account and has the correct username and password, then the scene will enter into the main dashboard.
* Build a Registration Screen – This screen is linked to from the Login Screen, allows the user to enter in a username, a desired password, a confirmation password, and try to register with the system. If the entries meets the system requirements, then the system will load the login screen again allowing the user to login to their new account.
* Build a User Account System
  + Keep track of basic user login information – This will hold the user login information and be the primary table to which all the other tables will link to the user key.
  + Keep track of user wins – This will be a table in the database used to keep track of the user’s track race placements. This is important to keep track of total net-income, game scores, etc.
  + Keep track of user expenditures – This will be a table in the database used to keep track of the user’s expenditures on custom in game items with in game generated money.
  + Keep track of user customizations – This will be a table in the database used to keep track of the users saved car customizations.
  + Keep track of the possible car options – This will be a table in the database used to keep track of the different customizations available for the cars.
* Build a Car Selection Screen – This screen is the interface that the user will have to customize their cars by selecting different wheel sizes, body colors, body shapes, and car weights. This will all be done through a sub sectional
* Build a Car Customizer
  + Customize Wheel Size – There will be a variety of wheel sizes for each of the cars. There are no effects that come from different wheel sizes, this is only for visual.
  + Customize Body Color – There will be a variety of body colors for each of the cars. There are no effects that come from different body colors, this is only for visual.
  + Customize Body Shape – There will be a variety of body shape for each of the cars. There are no effects that come from different body shape, this is only for visual.
  + Customize Car Weight – There will be a variety of car weights for each of the cars. The weight will affect how fast the cars will accelerate and how little the different terrain will affect the car.
* Build Track Selection – The following are done in parallel.
  + Different track lengths: In the tracks that the user can choose from, there will be different lengths to simulate different difficulties. The longer the track, the more difficult it will be.
  + Different track terrain: Tracks will have different terrains, some of which will allow the car to accelerate faster than default, decelerate the car, etc. The amount of terrain changes increases as the track increases.

**Risk**

This section will discuss the risks the user can encounter from utilizing the software and the risks our team can encounter from developing this project.

User Risks:

* Username and Password theft – Since the user is creating a private user account using a unique username and password, if another person (attacker) were to get ahold of that username and password, the attacker could go in and change the users custom car builds, or spend all the in-game money that the user hasn’t used on (maybe) unwanted purchasables. This is mitigated by the password field text being obfuscated with black dots instead of the keys.

However, since the user will be accessing the application locally, and all the saved information is in a local database, (sqlite) there is no network access needed, eliminating a large number of risks associated with network risks. Therefore, a lot of the risks experienced by the user outside of the ones listed above are arguably live on the system itself and indirectly affect our software.

Major Team Risks:

* Grades – If the project ends up being larger than anticipated, then sections of our grades will suffer. For example, under the project scope statement, if we do not have something running partially by 7/27 we can lose a substantial part of our grade, as this section is worth the most of our overall grade.
* Invalid System Access – If the software accesses the user’s system through an invalid method, our team will be liable for all damages and possibly culpable for other damages.

Minor Team Risks

* Unforeseen Individual System Bugs – If our software doesn’t run on systems, of whoms specs were not tested for. This could lead to outlier feedback that could hurt our teams credibility in small ways.

**Evaluation Criteria**

We plan to run a user study to evaluate usability

**Implementation**

* Build a Login Screen – Cory, Vishal
* Build a Registration Screen – Cory, Vishal
* Build a User Account System
  + Login Information Table – Cory
  + Race Metrics Table – Cory
  + Expenditures Table – Cory
  + Customizations – Cory
  + Options Reference Table – Cory
* Build a Car Selection Screen – Jerard
* Build a Car Customizer
  + Customize Wheel Size – Bill, Jerard
  + Customize Body Color – Bill, Seth
  + Customize Body Shape – Bill, Jerard
  + Customize Car Weight – Bill, Vishal
* Build Track Selection – The following are done in parallel.
  + Different track lengths - Vishal
  + Different track terrain – Seth

**Training**

This section will discuss the relative training a user will have to go through to be able to utilize our software effectively and then discuss the learning curve to move fluidly through the process.

The process flow of our software will be linear to reduce any confusion. Based on some of the answers given to us during the interviews, a lot of users remarked that an entry barrier to gaming and racing games was time and complexity. With a linear process flow, the interaction with the UI should be more intuitive and easy to follow. This will reduce extensive or repetitive training needed for the user.

Login Main Splash Car Customization Track Selection Race Screen

\* Options

\* Options Include:

1. Race Again
2. New Track
3. New Car Customization
4. Quit Racing

The Selection Menus will also be built to accommodate for simplicity. Each customization will be in the form of a roller with the options progressing from left to right. This means the user will organically move the arrows up/down to select different options within the header category, and left/right to switch category. Once the user has moved passed the final customization category, (to the right) they are able to move to the next screen.

Finally, the controls for the car during a race will be the left/right/up/down arrows between the keyboards main key block and number pad. They will all translate to car commands based on the intuitive interpretation. (I.E. up commands the car to accelerate forward, back commands the car to accelerate in reverse, left commands the car to turn left, and right commands the car to turn right)

There is no other training required passed this to understand the basics of our software, and the game can be played on this minimal amount of information. For extras, there can be training on purchasing custom car modifications and how weight affects the users car during a race.

**Maintenance**

This section will discuss the maintenance of the software post release. It will outline the boundaries that the software will operate in, and the access the users will have to install the software.

As discussed earlier, this software will be offline. This means there will be no servers, no call centers to answer questions centered around connection issues, and no scripts that will continually monitor network connections. Therefore, there will be no maintenance in that area.

Also, the software will be all encompassed in one software bundle, so there wont be any extra software needed to be hosted nor downloaded by the user. Again, this saves on the servers, call centers, and subscription software to keep the distribution going.

Finally, the software will be uploaded and distributed through Steam, a software distribution platform. Since it will be free, we will not have to maintain any account for monetary collection. Also, Steam maintains itself, so we may be giving away some terms under the license to allow Steam to host our software, but the license is open source.

The only maintenance needed is the retention of a lawyer for just incase another party decides to sue our company.

**Future Needs**

This section will discuss the future direction that this project can take. During our interviews, we were informed on a wide variety of future features that the program could have. The biggest feature that was discussed was multiplayer. Our interviewees almost unanimously had comments about wanting some different flavor of multiplayer content. These included racing against other people, interacting with them while racing either through in-game objects or collision repercussions, etc. The next feature widely mentioned was graphics. Given enough time, our next feature rollout would be updated model meshes to enhance the visuals that come with beautiful colors and sharp visuals. Finally, an indirect future feature mentioned was online multiplayer and scoreboard sharing. This would mean that the users would be able to race and compare scores all online or some local area network.