The **description** should provide a high-level overview of the project, compare it to any software programs already in existence (if applicable), list its most important features, and discuss the hardware and software it requires. The description should also indicate who the end user of your system would be, and, if relevant, how it would be maintained. A more detailed description of what you plan to accomplish will be included in Section 5 of the SDL, the Requirements Specification section, so only the top-level overview of the project should be included here.

The project that our group hopes to complete for this class is a robotic arm that uses computer vision to perform some kind of task. As of right now, we don't have a concrete idea for what the task might be. Some possibilities include a sorter that removes recycling or compost from landfill-bound trash, an arm that recognizes facial structure and attempts to feed the user, or a desk organizer that recognizes common items and organizes them on your desk. We intend to use additive manufacturing to manage hardware costs, which would require some simple CAD knowledge and access to the EDC. Right now, we imagine using a rPi hooked up to some servo motors to drive the arm. The easiest way to implement the software would likely be using python to control the GPIO pins of the rPi based on some camera input, likely an inexpensive webcam. For the camera processing, we would probably have to learn OpenCV, TensorFlow, or some similar libraries for object/face recognition. Another key feature that we would need to both learn and implement is inverse kinematics. In order for our robotic arm to move to the locations we need it to, we would likely use numpy or some similar math library to implement any inverse kinematics.

The justification should explain why your team's choice of project is appropriate for this class. The following criteria will be used to judge the appropriateness of the project, and should be considered in writing the justification:

- The project applies and demonstrates what team members have learned in other classes
- The project gives each team member the opportunity to extend what s/he has learned in other classes
- The technical difficulty of the project is appropriate (not too easy, not too difficult)
- One semester is a reasonable amount of time to develop the project
- The team as a whole possesses the tools required to complete the project
- The project is interesting to each teammate, classmates, and the instructor

We think this project is appropriate for this class due to its difficulty, opportunity to expand on topics we are interested in, and overall coolness. During our initial discussions on project topics, robotics came up quickly as a point of interest for us. Due to our focus on computer science, we hoped to find a project that would make use of interesting hardware and give us ample opportunity to write interesting software, without making the hardware unnecessarily complicated from a technical/EE perspective. We came up with a simple robotic

arm due to its versatility and relative simplicity. However, we still think the hardware will be a considerable challenge since none of us have extensive robotics experience. On the software side, we see this project as a great way to extend our knowledge from classes like AI, Operating Systems, and Algos. In addition this project will allow us to gain valuable experience working together on a multidisciplinary project that integrates software, hardware and theoretical knowledge. We plan to approach each aspect of the build together, to ensure that everyone in the group gains exposure to the hardware design, control software, and computer vision components. Through tackling challenges side by side, we will strengthen our ability to problem-solve as a team, which helps mirror the collaborative environment we're expecting to encounter in the industry.

Another reason this project is a strong fit for the class is its flexibility. The robotic arm gives us the option to start with a straightforward application, such as detecting and picking up basic objects, and then gradually increasing complexity as we make progress. This flexibility helps ensure that our project scope can expand or contract depending on our timeline, while still allowing us to demonstrate creativity as well as technical ambition. In addition, it also reduces the risk of becoming blocked at a single obstacle, since alternate applications or simplified goals can always remain viable.