CS 320
1.0 Release

Due: **Tuesday, December 10, 11:59PM (end of day) EST** via Canvas. For this assignment, you must work in your project group. Submit only one assignment per group. group. You will also give an **up to 10-minute** group presentation on Tuesday, December 10. Finally, you will submit a **video** advertisement of your product.

Overview

Release day is here! CSRocks Inc. can't wait to receive the delivery of your product. This assignment is the final milestone delivery. Your adoring fans, fame, and fortune await. Full milestone payment will be given for a product that meets its requirements, including reliability, extensibility, and maintainability. Evidence of good software practices during development is sure to convince CSRocks Inc. of these features.

As you complete your 1.0 release, you will present it to your customers and peers.

Deliverables

1. The 1.0 release of your product. The 1.0 release should show functionality targeted for the final release, as outlined in your SRS feature list, in place, integrated, and working, for all pieces of the system. Quality is important. The features in place should be solid and ready for use by a real customer base.

   The 1.0 release includes several elements packaged in three distributions (plus a video!):

   (a) Installed solution. (This distribution is only applicable to web apps and desktop applications. Mobile apps do not need installed solutions.) Please provide a url for a live version of your product. The customer will use this for the initial evaluation. This deliverable will be graded on whether it reflects substantial work and effort on the part of your team, has a solid and polished user experience, and successfully implements the features and use cases committed to in the SRS. Your product need not necessarily be 100% bug-free to receive full credit, but any known bugs must be backed by issue reports. Your system should be robust so that errors occur gracefully as much as possible.

   (b) Binary (installable) distribution includes the elements needed for an administrator to install your product on a server and run it remotely. For your web or android application, this may be a server package (zip file) or a url to invoke or android app to download. You do not need to package the database as part of the server for this assignment.

   The release should include clearly identified *release notes* in the package, describing (1) how to install and run the software, (2) which commands are working, and (3) a list of any known issues. Part of your grade will be on the clarity and quality of these instructions.

   Please assume the staff knows nothing about your platform and toolset and be very specific with the commands to execute to do the server installation. Please ensure the staff have access to your system database and your bug database.

   (c) Source distribution includes the elements needed by someone who is going to pick up the project at this stage and do further development. A separate package for each host target is often a clean way to organize this. Again, you do not need to describe the underlying database installation for this assignment.

   The source distribution should include clearly identified *release notes* in the package that describe how to build the product from the original sources. In grading, we will be looking for the ease with which a build can be made.

The release notes should state any assumptions, such as Visual Studio, Eclipse, or Android emulator installations. The release notes should also identify the source code repository and how to look at file change logs. A new developer may be looking at the change logs to understand why code evolved in a certain way, and we expect good developer documentation to be in place.

The release notes should also identify the issue-tracking database used in the project. We expect to see good usage of the issue-tracking database, especially for bugs that span development groups, by your team members.

Lastly, your software should be supported by tests, and a new developer will need instructions on how to run at least the following:

- unit tests for one major class or component in your system, and
- integration or system tests for the functionality represented by at least one use case.

Don't forget that tests need documentation, just like all code.

Again, please assume the developer knows **nothing** about your platform and toolset and be **very** specific with the commands to execute to build.

This source package will be graded on whether it demonstrates the following attributes:

- substantial work and effort by the development team,
- system design making use of design patterns, principles, and heuristics as appropriate (please use internal comments to identify these, and provide us with file names/line numbers of specific examples),
- adequacy of the Release Notes for the build process,
- ease at which a correct build can be made, and
- ease at which a developer could progress the system from this point (good internal documentation helps with this).

The test package will be graded on whether it includes:

- unit tests, integration/system tests, automation wherever possible,
- good coverage based on these tests (if you have data that indicates coverage, please include it),
- latest test results (a launching board for future development), and
- ease at which the tests can be understood and run by a new developer (instructions are necessary).

(d) Final admin and user documentation. The product documentation may be through manuals (admin and user) or online (user) help. The medium should positively impact the user experience. The document should cover all major areas of usage of the system and be both complete and of good quality.

2. An updated system requirements specification (SRS) and system design specification and plan (SDS) documents, with change tracking on (use as the baseline version, the β documents). Focus on the content changes, not the polish.

3. A measure of how well designed and extensible your system is, by identifying exactly what components must change, and how, to accommodate (1) interfacing your product with LinkedIn, (2) scaling it to handling 300 million users, and (3) identifying and removing or blocking bot and spam users.

4. A reflection of your top three lessons learned from the project experience. The lessons can be in terms of specifications, design, planning, teaming, execution, etc. What worked exceptionally well and what would you do differently next time?

Presentation

Your group presentation will be 10 minutes long. Your grade will be based in part on the presentation, so you should make sure you practice the presentation, and get outside feedback (from friends or classmates).

Remember: You are presenting to the customer! Keep technical jargon to a minimum, and present at a high level. Convince the customer that your project will be a success and that you are aware and ready for the risks, and have a well-laid-out plan. Make sure you start with a brief recap of the product — those who have not been working on the project as hard as you may not remember your goals.

The structure of the presentation is up to you, but including a live tool demo is always a great way to generate excitement for the product.

Video

To advertise your product, you will make a video: Your video should consist of two main parts:

- You should first outline the problem your product solves. Why would someone want to use it? What will it give me if I use it? Grab the potential customer and make them excited to use your product!
- Next, you should show a demo of your product, making it clear how easy it is to use the product, and illustrating its greatest and most exciting features. Not only do you have to grab the potential customer's attention long enough to watch your video, but you must also convince her to download and install your product. Make sure it's exciting.

There are many great examples of product videos online. Here are three examples:

- `http://www.youtube.com/watch?v=4KmsqYjB9j4`
- `http://www.youtube.com/watch?v=hmeZvp8U6EU`
- `http://www.youtube.com/watch?v=VFbYadm_mrw`

You will likely need to use video screen capture and voice recording software. There are some free options available online. Two example are `https://obsproject.com/` and `http://camstudio.org/`, but there are many others that may suite your needs better.

You may submit the video directly via Canvas, or submit a url to the video. You may use youtube or another video posting service if you'd like.

Submission and Grading

Remember that your product must be live on the url you give us when we grade it. It must live in that url until at least December 23, 2024 to allow us time to grade it.

As before, a small part of your grade comes from the looks or aesthetics of your documents. They do not need to be beautiful or excessively formatted, but developers and your customers need to be able to read them and extract information from them. This means they should be clearly written, with proper spelling and grammar, clear wording, and formatted with enough organization to present your ideas clearly to the reader.

Your 1.0 release should reflect customer interaction. If your customer is surprised by some features you have chosen to implement (or not to implement) or by a direction your project has gone, this will reflect poorly on your grade.

One of your team members should turn in all the deliverable material together so that there is one coordinated input for the team. Put the team name in the filename of all components submitted.