CSE 403, Project Phase 3a: Beta (60 points)
Second "Beta" Code Implementation

For the "Beta" release, you must produce a working initial version of your project, reflecting several (but not all) of the features listed in your SRS and SDS and expanding upon the partial functionality implemented in your "Alpha" version. Your Beta must implement many but not all of the overall features from your project.

Wiki Page for Beta:

Please create a **wiki page** on your GitHub that contains a direct link to your beta site or APK binary, any necessary instructions about how to run and test it, and a list of features that are/aren't implemented for this phase, along with any other information or instructions that would be required by the grader. Make sure to provide information about how to log in or access any authenticated resources related to your project as appropriate.

Binary Distribution:

The binary distribution contains the executable resources necessary to run and use your system. If it is a mobile application, your web site as created in the ZFR/Alpha phases should exist and should contain a link to download and run the product. If your system is web-based, its binary distribution consists of the site being up and running by the due date.

Your team will still be working on your project after turning in the Beta, but you should leave the Beta version available so the customer TA can test it. One way of achieving this is to put the Beta at a **separate Beta URL** or separate APK binary from your current version, so that one does not overwrite the other. When the customer TA goes to your Beta location, the binary or site there should be in beta form and not in the form of the latest version of the code that you have completed since the beta. Groups that fail to follow this separation may receive a substantial deduction.

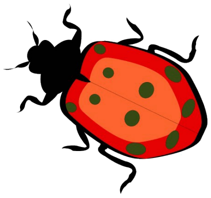
Required Functionality:

You will work on your project code in this Beta phase and in the following V1 phase. Each phase has roughly the same amount of time allotted to it. Therefore your group should meet and decide how the remaining features and functionality should be split up between the two phases. You should plan to implement roughly $^{2}/_{3}$ (66.7%) of the total functionality for your project during your Beta phase. (The V1 will implement the rest, plus bug fixes, testing, etc.)

In general, which $^{2}/_{3}$ are implemented is up to you, but we request that you follow these **constraints**:

- Target enough core functionality to implement all of your **use cases** and sequence diagrams from the SRS/SDS.
- Your Beta app should generally stand on its own and should implement a **coherent and complete** set of features that would make it usable in a live demo to a customer or group of peers. Choose entire features where the user can interact with the app and complete a goal or activity from start to finish, rather than a partial or incomplete feature. If you implemented many features but did not complete any of them, they would not be easily testable, so your project would not stand on its own.
- Your Beta must contain at least some **client/server functionality**; that is, some feature(s) that begins in your client app or site, then touches the server side or database side, then comes back to the client.
- Although your app will not be 100% complete, for the portion that is complete, you should present the user with a solid and **polished user experience** that reflects what you have learned from your UI prototype and Alpha.
- To make sure that you are targeting the right amount of functionality for this phase, **email your customer TA** ASAP during the first week of this phase (before your official in-person customer meeting) to come to an agreement with them about which features you will build. You may generally assume that the TA will approve a reasonable proposal, so you do not need to wait for their response to begin working; but the TA may ask to tweak the proposal to ensure that each group is completing an appropriate amount of work in the allotted time.

Bugs and Bug Tracking:



We expect that your group is updating your project's bug/issue-tracking system as you work on each phase. As we grade we will examine your bug-tracker to look for an adequate number of issues. Bugs should be thoroughly filled out with information such as: description, steps to reproduce the bug, severity, who the bug is assigned to, its current status (open, resolved, won't fix, etc.) Many bugs or issues might still be unresolved at this point in time; the main point is that some have been found.

We expect that your Beta may have minor issues that come up during testing. While the spec for the Alpha was tolerant of major bugs and missing features, the bugs and errors tolerated for the Beta phase are expected to be more minor. Any bug that remains in the Beta code should be a side issue, such as an incorrect message text or minor error message, and not a scenario-stopping bug that prevents the customer from being able to perform normal usage and evaluate your work.

Any known bugs or issues in your product should be documented using your group's **bug tracker**. If we encounter bugs during our own testing that are not represented in the bug tracker, you may lose points. We expect that several bugs will be present and filed in this system at the time your Beta code is due, including some bugs that have already been previously resolved/fixed and others that are still open.

Source Distribution:

Your project's source code, along with any other resources necessary to build and run your product, must be checked into your version control system by the due date of this phase. We will access your work through your version control system to grade your code, so this must be set up and reachable by your customers at any time. We expect you to be using your version control system properly, making small but coherent check-ins properly labeled with commit comments to indicate what has changed.

Code Reviews and Coding Standards:

Your group must continue to perform substantive **code reviews** over all non-trivial code check-ins for the project. Follow the same code review requirements and guidelines as outlined in the Alpha spec.

In the Alpha/SDS phase, you described your group's coding standards and how they will be enforced. We expect that your Beta code will follow these coding standards and that your code reviews will address violations of these standards.

Submission and Grading:

Submit your Beta code online by checking it into your version control system on **GitHub** by the due date. Your code review documentation might be on paper, in which case it can be given to your customer or instructor in person. We prefer that you would have some sort of digital artifact that can be stored in your version control system. Because you only see the customer/instructor in person at lectures and sections, you may need to slide any paper documents (such as code review reports) under the instructor's office door. This should be done by the due date stated in this document.

Remember that part of your grade comes from having a meaningful in-person **interaction with your customer TA** before the phase is due to show your progress, ask questions, get feedback, and generally make sure you are on the right track. Note that as described earlier in this document, you must ask your customer TA for approval of the functionality you will complete for Beta, in addition to (and prior to) the in-person meeting with the TA.

Your Beta binary will be graded on its functionality meeting the requirements described previously. Your Beta source code will be graded on its "**internal correctness**" as well. Specifically, we will evaluate whether it demonstrates substantial work and effort, is well-designed according to the object-oriented design heuristics we have learned in class, is documented using comments on each file and significant method or code section as appropriate, demonstrates general cleanliness and elegance of the code, and follows your group's style guidelines.

During grading we will access your work through your version control repository, so this should remain up and reachable by your customers at any time without prior notice. The Beta version of your code and resources must remain available and reachable even as your group advances to work on later phases.