Capstone Project: Documentation of Development and Assessment

Auction Team: Emily Clyde, Rob Callahan, and Ryan Recinos

1. **Revised Documentation located in PDF format in this zipped file named ‘*CapstoneFinalDocumentation.docx’***
2. **Evaluation of the development process**
   1. ***Communication with the client:*** *The client was very excited to be getting such a great team to work for them and put this application together. We met with the client a few times in person. Rob has been the go to guy as his son goes to the school as a student. They are not very technical, so we will be training them on how to use the system at a later point in the year.*
   2. ***Team Collaboration:*** *Our team was very good with collaboration. Before making any changes/revisions we communicated this to our other team members. We each played a part in the development of the system. This included coding, design, and project management.*
   3. ***Designing the system:*** *Designing the system was not too difficult. Though each member did both front end and back end design of the system. The most helpful thing during the development process was that we had a clear picture of the front end design before even starting the project.*
   4. ***Project management:*** *Project management was very easy for our team. We all tried to implement items into the system. There were a few bugs. Being a scrum master in this team was easy as each team member held themselves accountable for each task (even those tasks not assigned to them).*
3. **Evaluation of the Finished Project**
   1. ***Assessment of the system***
      1. ***Is the system done?*** *For the most part the system is done. If the system needs any polishing we plan to complete and update the system this summer.*
      2. ***Does it do what the client wants it to do?*** *Yes. The website is functional for an auction now.*
   2. **Assessment of the quality of the design:**
      1. ***Where there any features that were hard to implement because of the design?*** *The CRUD design in the packaged software made it easy to implement. All we did really was tweak some HTML, CSS, and a tiny bit of code. The hardest part was figuring out what we needed as we got deeper into creating features needed for the Administrator.*
      2. ***Will the design be easy to extend (or to add features)?*** *The design should be easy to extend and add features. Many of the pages are based on the same design (tables) so adding another page or feature would be easy to do.*
      3. ***Is the design easy to understand and communicate to others?*** *Yes. The design is straight forward. The main page during the auction event is the auction forms page. This is the engine of the auction site! Everything can be accessed on this page!*
      4. ***Does the design lead to simpler (lower cost) hosting or distribution requirements?*** *Yes. The database is very small and should remain under 20 MB. The database will have the ability to be cleaned after each auction event. The website will be run once a year for a few weeks at a time. The design was to make it so the client could keep costs down as they do not have a large budget to host a website. The client should be able to use the free version from Azure with not issues!*
      5. ***Does the design give sufficient performance?*** *Yes. We see no problems with older machines running this application at the current time.*
      6. ***In retrospect, would a different design be easier to implement?*** *No. The CRUD design given to us by the entity framework was the easiest to implement this project. All we had to do was tweak a few things in the design to get the results we wanted.*
   3. **Assessment of the implementation (coding) of the system:**
      1. ***Will the code be easy to maintain?*** *Yes. See comments below.*
         1. ***Is it easy to fix bugs?*** *Yes. There is not a lot to go wrong with this application! We have great documentation, code, and separation of concerns!*
         2. ***Will it be easy to add new features to the system?*** *Yes. We used the standard framework under the hood!*
         3. ***To what degree did you follow best coding practices?*** *We used separation of concerns in this project. We appropriately commented in controllers when needed.*
      2. ***Is the code well documented in the code internally?*** *Yes. See comments below.*
         1. ***Are there descriptive name for classes, methods and variables?*** *Yes, we used descriptive names based on the auction concept (i.e. bidder, auction items, donor, and contact) for many of our model (classes), controllers, methods, and variables. The project documentation has UML designs for reference.*
         2. ***Are there comments where needed to describe what the code is doing?*** *Yes. We made sure to put comments in our code. They can be found in the controllers as well as in the HTML pages.*
         3. ***Are there unit tests that make the purpose of the methods explicit?*** *No. We felt it was not necessary to use unit testing at this time. Much of our code is system generated from the entity framework so we are trusting that those methods already created are okay.*
      3. ***Will your code run fast?*** *Yes. The website is not that big, neither is the database. We have no images in the database as well, this will help the application run faster (especially on older slower machines).*
      4. ***Is your code efficient?*** *Yes. We have some methods we created but like we mentioned before we are relying on the system generated entity framework code to run the application.*
      5. ***How bug free is your code?*** *The website has been beta tested. It can be used now, if the auction event were to take place.*
      6. ***How do you handle exceptions?*** *We have not created any exceptions in the code at this current time. Anything that is in the code now is system generated. We may add exceptions at a later point in time if it becomes an issue. However, the site is only going to be “live” once a year for a few weeks at a time. It will not be live the entire year only for this one event.*

Overall Assessment of the Capstone Project

1. **Purpose of the project:** 
   1. *The purpose of the project was to create a web application that our client, Logos Christian Academy, could use during their yearly auction event.*
2. **All the features you have implemented:**
   1. *The features that our client wanted in this application are the following:*
      1. *Ability to promote the Auction before the event*
      2. *Ability to Add/Edit Bidders, Items, Donors before and during the event*
      3. *Ability to save and print reports relating to the auction event*
3. **Features the client wanted but that haven't been implemented:**
   1. *Unknown until we run it for the first Auction Event! We have implemented what they think they need right now.*
4. **The feature that you take the most pride in:**
   1. *There are many! Figuring out how to print reports! We decided on using an open source third party application that allows the user to save for later as well as print reports. Integrating other technology can be tricky but this works pretty well! Loading images into files rather than into the database as well. Our view models are pretty great as well!*
5. **The part of the project that was hardest:**
   1. *Identity! Hands down! Though it was pretty easy to link the tables related to identity to our database.*
6. **The feature that you most wish you had more time to work on:**
   1. *Identity. There are some tutorials that created a view page to assign roles, users to roles, and delete users. There was not enough time to implement these features! So we defaulted to one admin user for identity.*
7. **The most important thing you learned while doing this project:** 
   1. *There is no such thing as one answer. There are many ways to implement features. Finding the simplest feature to use is far harder than anything else in this project!*
   2. *GITHUB high learning curve here! Getting better at it but it can be testy sometimes.*
   3. *Keeping back-ups is critical! Glad we did that!*
8. **The one thing you will do differently on your next project:**
   1. *When a feature is working do not change it! And test, and then test again, and then test it again! It is a cardinal rule not to test your own code, so we needed to test more during the project!*