**Question 1:**

1.) Synthesize learning through critical analysis: My contributions were primarily to the creation of the business layer. The creation of an application flow diagram, model and service prototypes, and managing team members and their tasks with the appropriate pre-conditions and post-conditions were all part of my primary role as a scrum master. Concepts I was directly involved with include refactoring code to optimize our apps performance, using the appropriate data structures to store information in our models, use of the observer design pattern to dynamically update the view from model state changes, design functions that through function composition preform all of the business logic for the application. Along the way, I learned from my colleagues and became better at object oriented approaches, using tools/techniques such as dependency injection, inversion of control, frameworks (MVVM), and using packages to decrease our overhead.

2.) Apply theoretical constructs to practical approaches: The structures I had experience with includes modules such as the Models, Services in our autofac container, the use of these modules in an application framework (MVVM), the use of design patterns such as the observer pattern which monitored our Models for state changes, and the creation of our own data types to store information efficiently in our Models.

3.) Critique philosophical tenets and current practices: There are many different issues we encountered whilst in development. However, as a team, we were able to create solutions and work around them. These issues represent ones that we will likely encounter in the field. Issues such as a problematic team member and negotiating with that team member despite their lack of involvement in the project, dealing with conflicting design decisions and treating these conflicts as productive and not personal resulting in better code quality overall, dealing with information security and privacy especially with a networked application like this one where unethical use could lead to breaching of privacy we never intended for our users, creating a flexible and high quality product for a client whilst negotiating design decisions with him. The use of frameworks and design patterns also improves the readability of our code and the ability to transfer knowledge and I am just now finally seeing the usefulness of these design principles.

4.) Integrate and refine oral and/or written communication skills: Communication was the single most important part of this semester without a doubt. Direction was the most difficult part of this semester, and towards the beginning, we lacked direction and didn’t really know each other as a development team. As time went on, we created sub-teams and positioned leaders into each team to facilitate unified design decisions and centralize communication throughout the team. I was initially the leader and scrum master of my team, the business layer team, and I helped create tasks and the overall design of the business layer’s services and models. I then translated these schematics to my team members. After the creation of our prototype, the need for such a leader diminished, and the business layer accepted a few new members and lost few others in a team reorganization. During the refactor and polishing stage of the business layer, our openness to critique each other’s code led to a complete restructure of the entire layer, but our code quality skyrocketed. What I’ve observed is that not every sub-team had this productivity and I attribute it to the great communication and centralized leadership we had on some teams but not others. Two of the three layers are complete (Data transmission and Business), but the UI team never had the manpower or advantages of this communication and it shows (lack of leadership, absent team member, no critique or group design meeting). It is in this class that I’ve been able to personally see the advantages of great communication.

5.) Verify our expertise: For a long time, I wondered how I would get from writing “Hello World” to creating entire applications. But now I can see all the moving pieces coming together and a beautiful machine working with all the cogs underneath. Working with the MVVM framework, creation of modules, using design patterns, choosing the correct data structures, understanding function composition, pre and post conditions, high level application diagrams, special techniques such as dependency injection and inversion of control. These are all concepts that I had learned about but it was not until this class that I finally got to see how important each part is and how they all work together.

**Question 2:**

The grade I believe I should receive is a B. Although I believe that my contributions were necessary to the teams overall progress, I cannot say that my contributions were anything close to the top members of this team who truly went above and beyond. My contributions were as follows:

1. Research and development of our applications network capabilities. A lot of time went into this stage of development but early on, almost all of our time was research finding our direction. I personally researched countless solutions for cross platform communication, but exhausted almost all of our options and we chose to take different design decisions later (server based model).
2. After this initial stage, I was promoted to be in a leadership role as a scrum master/product manager for business layer. I designed the application’s functions in the services, their pre and post conditions, and managed two other team members and helped them create the various modules I required for my original design to function. I organized our priority schedule and communicated with the two other teams. I created design schematics for these members to follow and although this position was intended to be temporary, many member advocated that I should remain in the position due to the spike in productivity. I helped create the services as well, but my team members would likely say that my biggest role at this stage of development was unifying our sub team, removing blockers, and reviewing code for quality assurance.
3. After this intermediate stage of development was over, the need for a leadership position faded and a new refactor/polishing team was assembled out of some old members and new members. Almost all of the code was rewritten during this stage of development. My overall design still exists, but we changed our approach in almost every way. I helped with this refactored and was mentored by other members on the most efficient design strategies. I assisted with the changing of our data structures to be our own data types and utilize the LINQ package to emulate dictionaries with less overhead, the implementation of the observables with model collections to monitor for state-based changes, the use of enumerations to create “armament” types, implementation of missing mechanics that we missed and the refactor of mechanics to be dynamic. Later down the road, these design decisions will help the expandability of the code. I also helped organize the two way binding between the view and the business layer. However, communication issues and lack of design progress in the UI made it difficult to connect the layers. As I mentioned before, maintaining code quality was of the up most importance at this stage and learning that a refactor early will save infinitely more time later was more evident than ever. Quality assurance was also of upmost importance, and I helped participate in the design meetings where every member was critiqued on their code and the openness to this led to the huge spike in productivity and high quality code.
4. Some miscellaneous contributions include: working on the presentations, managing the trello, helping members research and understand design concepts or packages, help setting up development environments.

**Question 3:**

1. Brandon: The indisputably most experienced member, Brandon’s contributions to the project made the entire thing possible. Without his expertise, leadership, general knowledge of software design principles, and amazing communication skills, the team would likely be nowhere near as productive as it was. His visions are visible throughout the entire project and he helped guide every single team member in times of struggle.
2. Ken: Ken was Brandon’s right hand man and his contributions dwarf ever other member of this team. His contribution to the data layer along with Brandon’s helped them single handedly created the Data Transmission layer despite the lack of manpower. Ken later joined the Business Layer refactor team and his updates to the layer and his ability to transfer knowledge from Brandon to members like Gerom and I helped the productivity of the layer peak. His communication skills and coding ability are also huge advantages to why he was such a great team member.
3. Joe: Joe was also extremely beneficial to the team. Like the others, his communication skills were amazing, he was great to work with, and his coding knowledge was fantastic. His role changed the most throughout the development cycle, but his role was to switch to any position that required extra assistance. He helped in almost every layer at some point and made a great general purpose developer that boosted productivity in all locations.
4. Josh: Placing myself was difficult, but I am placing myself here because of my role as a leader during a critical period in our development. I believe that the top three people on this team did significantly more work than myself and the others below my ranking, but I believe that my contributions to the business layers creation, polishing, and my assistance to multiple team members warrant my position here. Honestly, positions 4-6 are pretty close to being tied for contributions.
5. Gerom: Gerom was my right hand man for the entire semester and he assisted me with everything from the research in the beginning, the creation of the business layer services and models, the prototype of the application diagram, and the polishing during the refactor team’s existence at the end of the semester. He was always willing to help and his contributions are in line with John and I.
6. John: John was the design lead for the UI team and he helped design the entire UI with detailed schematics. Were it not for the lack of manpower on his team, the UI would have likely been finished as well. However, John may not have need the best for a leadership role. His other classes interfered with his ability to communicate and unlike the other layers, he did not have any design meetings or group peer review sessions. From interviews with members from the team, there seemed to be lack of communication across the board. However, John’s contributions to the presentations and miscellaneous tasks can’t be ignored either. His contributions are still significant and in line with Gerom and I.
7. Emily: Emily was generally responsible for the communication between team members and the client. Her biggest peak in productivity was when I was managing her while she was on the business layer team. She helped created the services, their functions, and she did it all autonomously based on design schematics that I had provided for her. However, after the prototype’s creation, she was voted to be sent to the UI layer team, but subsequently had no guidance or leadership and her contributions suffered as such. She also had a very packed schedule outside of class, so I can’t really fault her too much for it. However, from a strictly analytical view, her contributions were less than the rest of the team’s.
8. Justin: Honestly not much to say here, he showed up to meeting sometimes and that’s about it. I think the “contributions” he made to the team are obvious.