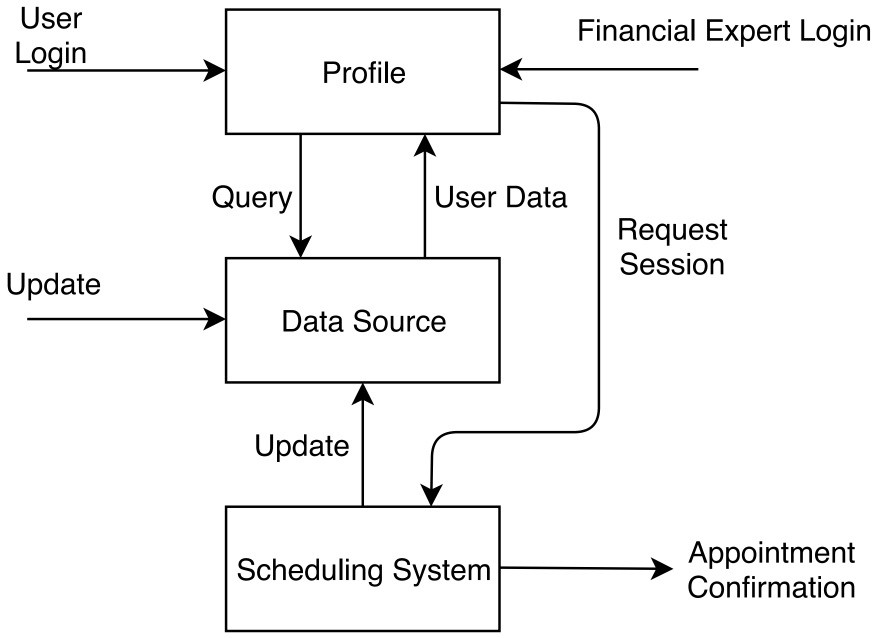
CSCI577a - Fall 2019 Isaac Delgado
HW1: ICSM USC ID: 4189182267

For the upcoming features to be developed or assuming you may redo your project with no schedule constraint, you may improve your project by simplifying or complicating (advancing) your project requirements/ features.

1. *(3 points) Draw a block diagram explaining overview of your project, or specific module that you want to simplify or complicate*
2. *(9 points) Provide 3 Developer simplifiers with supporting rationale*
3. *(9 points) Provide 3 Developer complicators with supporting rationale*
4. *(9 points) For each simplifier and complicators mentioned in 2) and 3), briefly explain about its risks and trade-offs.*

1.) Scheduling Module



2,3,4)

Simplifier	Rational	Risks & TradeOff
Reuse UI/UX for both types of users	This will be a simplifier as we will not have to re-develop a completely different UI for each type of user as we can re-use the initial workflow and simply give access or restrictions depending on the type of user.	The risks associated with this is we could be lacking on the quality of the UI/UX for any of the types of users. Additionally if client decides to add big updates to one of the types of users then we may have to completely develop their UI/UX at that point.
Financial Experts will pick key words from a list that we provide to let them be search for by those words as key-words or categories	Instead developing a search algorithm to define financial experts under key-words and categories, we can simply let the financial experts choose their own categories and then give category options to regular users which will then fetch the list of financial experts that choose that key word. This will allow for a simpler search algorithm of just key word matching.	This could make it harder to find specific financial experts especially when the list of financial experts is large. Developing a more sophisticated search algorithm could help connect regular users to a perfectly matched financial expert.
Using a scheduling API, such as Nylas, to take care of scheduling.	Scheduling is a very difficult task to program well. Instead of developing the scheduling system from scratch we could try to use a scheduling COT and embed it into our application.	We do not know how compatible some of these scheduling COTS are especially with a mobile application. This could also potentially make the system slow as this might mean we will have to send data out to the COT and get data back.

Complicator	Rational	Risks & TradeOff
Develop two separate UI/UX for both types of users.	Developing two systems for the regular user and the financial expert can allow for better quality and control for each user. This is similar to how Uber drivers and a separate system from the Uber customers. This will complicate the system as a whole new design is required for each user.	The risk is the potential of running out of time. With our course we have very limited time and developing completely new systems for both users means more work to be done. Additionally all the risk that comes with more development is included like personal shortfalls.

Complicator	Rational	Risks & TradeOff
Developing a financial expert search algorithm by scraping financial expert profile pages and history of sessions.	This will allow for a more accurate search for financial experts as this will learn who the experts are based on their page and feedback from customers. When users begin to search they can type any key words the search algorithm will index through profile pages of financial experts	No one on the team has developed a search algorithm to this extent before. So the risk associated is personal shortfall with regards to skills and experience. Thus could possibly push the project past the allotted timeline if we tried to develop a complicated search algorithm.
Develop the scheduling system from scratch.	This will allow for more control over how the system will handle the scheduling functions. We will also have no trouble integrating it into our mobile system as we will use the same language as the mobile application, in this case react native.	Developing a scheduling system is difficult. There are many scenarios to take care of when developing a scheduling system. This may be out of the scope of the timeline given the team’s experience and time dedicated to the project.