**Why Amazon?**

First, I have always dreamed of developing products that can make people’s lives easier, that can enrich people’s lives. Amazon has a strong focus on customer experience and has built many such products. To name a few, the retail site, prime videos, Kindle, Alexa and so on. Those products have reached the broadest range of normal people's lives. It would be very rewarding to use my skills to build such products. There is no better place that can fulfill my dream to develop products that can impact people’s life for better.

Secondly, I always wanted to work with large scale applications. Considering amazon customer base, every application at amazon is large scale. Although large scale applications can be very challenging, they always fascinate me. They can push my limits. There will be many tough problems coming out of such applications, I will have to figure out smart and innovative way to solve those problems. My moto is work with talented people on challenging things. Amazon has so many talented people creating amazing products every day. I will never get bored working at Amazon.

So, amazon is the ideal place for me, where I can use my skills and experience to fulfill my dream and learn a lot.

**Tell me about yourself**

My name is Min Xiao. I’ve been working on professional software development for about five years. During the five years, I have developed a few web applications and web services mainly using .Net tech stack. Those projects were well received by stakeholders and end customers for great performance and user experience.

My experience in those projects includes communicating and analyzing requirements, designing workflows, building prototypes, implementation, configuration and deployment, which covers full software development life cycle. I have been working full stack on those projects, which is good for me to understand an application from end to end. I am now very interested in Microservice for the many benefits it brings. Similar to Microservice, I am also quite interested in distributed systems.

I’m passionate about learning new technologies and incorporating them with what I’ve been doing. The new technologies could refresh my mind with better designs and solutions. I am now looking for a place to use my skills and experience to build great products and continue to grow.

**What's your strength / weakness**

I have strong logical thinking ability in solving complex problems, which is an essential skill for a programmer, and that's why I am here.

I am a very resilient person, that is an important quality when dealing with tough problems. When my 10th approach failed, I will come up with 11th approach and try it if the problem must be solved.

I pay great attention to details and always make sure things are 100 percent completed in my hands.

Speaking of my weakness, it may be my public speaking. I am more of an introvert person and used to be pretty shy. But I know it is an important skill to advance my career. So, I am trying to improve it now. I have joined local ToastMaster club to practice my public speaking.

**Tell me about a time you failed and what you learned from it**

When I was working on the deduction service for the onboarding project, everything worked fine during development. After the service went live, customer reported many errors pointing to that service, which was strange to me at first. So I dug into the logs and source code, after thorough investigation, I identified the root cause. One table in database changed to a different name, but I forgot to update the ORM object associated with that table. In lower environment, somehow both the old table and new table existed, so the services worked fine during Testing in lower environments. I immediately updated the ORM object and pushed a hotfix to correct the error. After my fix was deployed, everything worked fine. It was definitely a silly mistake. I apologized to my manager and the customer.

So, after that, I learnt that I should pay attention to each small detail during the development process, and document them in time so that I won’t forget. I need to make sure that all the things I delivered are fully qualified. [2019, June]

**Tell me about a time when you were not able to meet a time commitment. What prevented you from meeting it? What was the outcome and what did you learn from it**

I remember last time when I was developing debit account change feature for our main payroll product. The original timeline was passed before the feature could be fully delivered. Main reason is on my side, I underestimated the complexity of the feature in the beginning and gave a shorter estimate to stakeholders, which became set schedule. Bank account is sensitive info and additional security had to be incorporated for the feature. I need to integrate with External verification system, GIAC. The integration took me more time to get it working. Second reason, stakeholders were asking for adding an internal tool to monitor and manage all bank account changes. This change came to me late in the project, as Agile team, while I welcome changes, but additional time needs to be taken into considerations.

Knowing that original timeline was not possible, I worked with product owner and came up with alternative plan to do two releases. First releases include all the customer facing piece and the required security piece. Second release includes the internal tool. After explained the plan to stakeholders, they showed understanding and we proceeded that way.

From this experience, I learnt I need to consider all aspects of a project and give more accurate estimate. Also, I learnt delivering useful feature for customers fast is important, I should make all possible ways to achieve that. [2018, Dec]

**We all deal with difficult customers from time to time. Tell me about a challenging client-facing situation and how you handled it**

When I was developing the e Disclosure application for UCSD conflict of interest office, one important piece of the project is to define states and design state transitions. The COI office people used to handle disclosures on a manual basis. There was no clear and defined workflow at that time. It was difficult to get started at first.

In order to get it work out, I collected and read all related documents I could find. With a decent understanding of their current manual process, I identified all the states and designed state transitions. I drew workflow diagrams for different type of forms, then had several meetings with stakeholders. I explained each state on every workflow diagram and how state transition worked, what would happen during state transition. After several rounds of meetings, they fully understood the workflows and even appreciated my design, which made their work more efficient as I did cut out unnecessary steps. This piece laid the foundation for this project. When the project was released, both regular end users and COI admin users liked it because of the simplified process. [2017, Jan]

**Tell me about a time when you had to push back to HQ or challenged a decision**

I remember when I was working on the earnings feature for the new enrollment application. There were different opinions on the workflow. The stakeholders decided to put the earnings view inside the employee module and guarded by a condition. I disagreed with the decision. That workflow would only hide the view and it was not logic to put company earnings on each employee level. So, I challenged the decision. I did new design and wrote a prototype, called out meeting to demo my work. They bought the new design. I committed to the new design and did implementation based on it. When the feature was released, it went really well, the experience was so smooth and intuitive for customers that the percentage of customers using this feature increased a lot compared to the old enrollment application. [2019, April]

**Describe a time when you sacrificed short term goals for long term success**

When I was developing the back end microservices for our new enrollment application. In the beginning, I had two option on database part for those services. One option, I could use existing main database, which is less complex to implement would take less time. The other option is to spin a new database, which would take more time to implement, and deploy. I weighed between the two approach. The second approach had the risk of missing the timeline. But in the long run, the new database approach would separate concerns, make the services loosely coupled with our main product system. We could independently scale the new database without worrying about other systems. Considering the benefits the second approach would bring in the long run. I decided to use a new database for my services. I did spend more time on the development because of this choice. But I made some extra efforts and in the end I was able to deliver before the timeline. [2019, Jan]

**Tell me about a time you stepped up into a leadership role**

When I was working on the onboarding project, in the middle of project, the team lead left our company. At first, all team members were, worrying about the ongoing project. I was one of them too. But after a short time, I figured it was my project too. I had put so much effort on it and I want the project to be successful and I believe everybody else in my team would think same way. So instead of waiting for the executives to assign new resources, I stepped out and took the responsibility to lead the project. First thing and most important thing at that time is to motivate my team. So in the next team meeting, I specifically called out that I would like to lead the project forward together with team, I spoke out my thoughts and shared that everyone here wants to make this a successful projects and we don’t give up. I am not a great speaker, but sometimes the action, the attitude, and the determination are more effective in motivating people. What I did during the rest of project and the determination I showed truly motivated the group. Gradually, the team morale was back and we delivered this project successfully. [2019, July]

**What’s the most innovative new idea that you have implemented?**

When I was working on the validation part for the new enrollment application. After I reviewed our historical enrollment data with our data analyst, I found the previous way of validating on each page in our old onboarding platform was causing customers to drop and negatively affected the completion rate. As customers may not have all the info at the moment but still want to move to next page. To avoid this drop, I had to come up with a new validation strategy. I thought why not remove the validation on each page. After further investigation on the techniques of Angular Single page application and state management, I further confirmed my innovation idea to remove validation on each view so that customer can move forward and back as they wish. They only need to correct validation errors on a centralized place when they were about to submit. This new approach of validation has significantly improved user experience. The completion rate has increased more than 100% compared with old platform because of the smooth user experience. [2019, Feb]

**Tell me about the toughest decision you've had to make in the past six months**

I remember when I was developing the back end microservices for our new enrollment application. I had two option on database part for those services. One option, I could use existing main database, which is less complex to implement. I would be able to meet timeline for sure using this approach. The other option is to spin a new database, which would take more time to implement, and deploy, and had the risk of missing the timeline. I weighed between the two approaches. Although using new database would bring extra work and may miss timeline, in the long run, it would separate concerns, make the services loosely coupled with our main product system. We could independently scale the new database without worrying about other systems. Considering those benefits in the long run. I decided to use a new database for my services. It was a tough decision because of the risk. I went on with the second approach of new db, it worked really well and I had seen the benefits of isolated new database. [2019, Oct]

**Tell me about a time you had to deal with ambiguity**

I remember when I was working on the workflow for e disclosure application at UCSD. I need to define the states and design state transitions. The conflict of interest office people used to handle disclosures on manual basis. There was no clear and defined workflow at that time. What they want is just a more efficient system. It was really vague at the beginning.

In order to get it work out, I collected and read all related documents I could find. With a decent understanding of their current manual process, I identified all the distinct states and designed state transitions. I drew workflow diagrams for different type of funds and forms, then had several meetings with the stakeholders. I explained each state on every workflow diagram and how state transition worked, what would happen during state transition. After several rounds of clarifying, updating and explaining, both sides were clear and satisfactory with the workflow. The stakeholder even appreciated my design, which made their work more efficient as I did cut out unnecessary steps. This piece laid the foundation for that project. When the project was released, both regular end users and COI admin users liked it because of the simplified process. [2017, Feb]

**Tell me about a time when you were 75% through a project, & you had to pivot strategy**

When I was working on Enrollment project at Paychex. I was more than halfway through the project. For the validation piece, I focused on putting validation on each view, and I had implemented the validation for most of the views. But after I read some historical enrollment data on old platform, I found the preventing way of validation on each page was causing customers to drop and that had affected completion rates a lot. Customers may not have the necessary documents or info at the moment. To avoid this drop, I had to come up with a new validation strategy. After thorough thinking about Angular application and Redux, I came up with the idea to put the validations in a centralized place. So customer can move forward and back as they wish. They only need to correct validation errors when they are about to submit. Then they just go to the fix validation error page and correct the errors. This change of approach has greatly improved user experience. The completion rate has increased more than 100% percent (30 to 60%) compared with old one because of the smooth experience. [2019, August]

**A time when you faced a problem that had multiple possible solutions**

I remember when I was developing the Company service for the new onboarding application. This is the first service developed to support the single page application front end. For the database part of this service, there were two solutions, one was to use the exiting main database, the other option is to spin a new database. Both would work and have its own advantages. I had meetings with other engineers and dba, to discuss over which approach to take. But We did not find which approach was necessarily better.

So I said that unless we had any new info, I’d rather we just pick one and move on. I picked new database option, and moved our development work. [2019, Feb]

**Tell me about a time when you missed an obvious solution to a problem**

When I was developing the PDF generation piece for the remote check printing project. After I did all the design, I started to code for my solution, then one of my colleagues mentioned that in our system there was a similar PDF generation functionality already. You probably want to utilize the existing one. I was shocked that I missed such a obvious solution to my problem. So instead of creating duplicate code, I looked into the existing PDF generation code, found it would meet my project needs with slight modification. In order to promote code reuse, I refactored that code a little to make it generic and created a service for it so it could be used by other people and other systems in the future. My project used my Pdf generation service well and later on it was used by several more projects from other group.

From this experience, I learnt that when solving a problem before I go directly and develop a new solution, I should check if there is existing code that I can utilize to solve my problem so that I don’t develop duplicate functionality and be more efficient. [2018, May]

**Tell me about a time when you received negative feedback from your manager. How did you respond?**

I remember there was a time in my 1on1 with my manager. He brought up that my demonstration in sprint review was too rush and wasn't clear enough. My first feeling was a little frustration. I said sorry to hear that, I will work on that. After the meeting, I reflected on the feedback [and figured that] I could have tried more on this demonstration thing.] In my mind, I used being a foreigner as an excuse for myself [not pushing on those areas]. I figured out there is more solutions than the challenges. I could do more than I used to think. After that, I always did more planning ahead. To prepare the data, the workflow, or have a script. during the demo, I tried to slow down as I have accent, also it is beneficial for audience to allow them to follow. To improve public speaking, in my private time, I attended speech club, such as toaster master club to practice. During standup, I also tried to articulate what I have done, the issues I encountered as best as I could. After a period of time, I did see people appreciate my demo. My manager did praise me for putting the efforts to improve my demo. The whole thing is rewarding for me too, to see a better version of myself. [2019, June]

**Describe a situation when you negotiated with others in your organization to reach an agreement.**

I remember when I was working on the validation framework for our new Enrollment application. The product manager and I had different opinions on how to do validations. His idea was to do validation on each view and if there was error the user cannot move forward. My opinion is that the customer may not have access to all the documents at the time, but he should be able to move to other views and enter more info within that session, thus we should use a non-preventive way. At first, the product manager did not buy my idea. So I called a meeting with the product manager and one data analyst, from the user enrollment data on our old platform, it is found the preventing way of validation on each page [in our old onboarding platform] is causing customers to drop and that had affected conversion rates a lot. I also ensured the product manger that I will do a centralized validation before sign and submission so that the required info was entered, and verified. Afterwards, he agreed on my idea.

My approach has brought a smooth experience for the new enrollment app and increased completion rate more than 100% percent. [2019, Feb]

**Describe a long-term project that you managed. How did you keep everything moving along in a timely manner?**

I would like to talk about the Enrollment application. This application is used for new customer onboarding, it is to replace the old enrollment platform. I have worked on it since beginning of 2019, and I'm continuing enhancing it. From the start of this project, I had gathered requirements from business, worked with UX team, and done initial release planning. As I was developing this project, every week I had project status meeting with stake holders and team members. Every two weeks, there will be a demo for the work completed. I had two phases for the project, phase 1 was to release it to end customer within half a year with essential functionalities. Phase 2 build additional features based on our Payroll products. By breaking the big project into small, measurable goals, I was able to move this project in a timely fashion. [2019, Jan]

**The last time you had to apologize to someone**

When I was developing the deduction service for the onboarding project, everything worked fine during development. After the service went live, customers reported many errors pointing to that service, which was strange to me at first. So I dug into the logs and source code, after thorough investigation, I identified the root cause. One table in database changed to a different name, but I forgot to update the ORM object associated with that table. It was definitely a silly mistake. I sincerely apologized to my manager and business people. I explained the cause and the plan to fix the errors to them. They showed understanding and accepted my apology.

I then immediately updated the ORM object and pushed a hotfix to correct the error. After my fix was deployed, everything worked fine. after that, I learnt that I should pay attention to each small detail during the development process, and document them in time so that I won’t forget. I need to make sure that all the things I delivered are fully qualified. [2019, May]

**Give me an example of a time you faced a conflict while working on a team. How did you handle that?**

I remember when I was working on the validation framework for our new Enrollment application. The product manager and I had different opinions on how to do validations. His idea was to do validation on each view and if there is error the user cannot move forward. My opinion is that the customer may not have access to all the documents at the time [he input on that view], but he should be able to move to other views and enter more info within that session, thus we should use a non-preventive way. At first, the product manager did not buy my idea. So I called a meeting with the product manager and one data analyst, from the user enrollment data on our old platform, it is found the preventing way of validation on each page was causing customers to drop and that had affected completion rates a lot. I also ensured the product manager that I will do a centralized validation before sign and submission so that the required info was entered, and verified. Afterwards, he agreed on my idea.

My approach has brought a smooth experience for the new enrollment app and increased completion rate more than 100 percent. [2019, Feb]

**What did you do when you needed to motivate a group of individuals?**

When I was working on the onboarding project, in the middle of project, the team lead left our company. At first, all team members morale was down, worrying about the ongoing projects. I was one of them too. But after a short time, I figured it was my project too. I had put so much effort on it and I want the project to be successful and I believe everybody else in my team would think same way too. So instead of waiting for the executives to assign new resources, I stepped out and took the responsibility to lead the project. First thing and most important thing at that time is to motivate my team. So in the next team meeting, I specifically called out that I would like to lead the project forward together with team, I spoke out my thoughts and shared that everyone here wants to make this a successful projects and we don’t give up. I am not a great speaker, but sometimes the action, the attitude, and the determination is more contagious and effective in motivating people. What I did during the rest of project and the determination I showed truly motivated the group. Gradually, the team morale was back and we delivered this project successfully.[2019, August]

**Tell me about a time you had to quickly adjust your work priorities to meet changing demands.**

When I was working on validation framework for the onboarding project at Paychex. Originally I focused on putting validation on each view, and I had implemented the validation for some of the views. But after I had some meetings with data analysts and stakeholders, I found the preventing way of validation on each page in our old onboarding platform was causing customers to drop and that had affected completion rates a lot. Customers may not have the necessary documents or info at the moment. To avoid this drop, the best way for me to do is to switch to a non preventive way of validation. So instead of validating at each step and preventing from customers moving to next step when there is error, I came up with the idea to put the validations in a centralized place so that customer can move forward and back as they wish. They only need to correct validation errors when they are about to submit. This change of approach has greatly improved user experience for the new onboarding application. The completion rate has increased more than 100% percent (30 to 60%) compared with old one because of the smooth experience.