

## About Myself

- My name is Jeremy, and I previously wrote code to increase fertility success rates nationwide
- As a Software Engineer at Oma Robotics, I used C++ to do two things: (1) implement user interface features to assist fertility doctors with medical procedures and (2) develop robotic algorithms in order to fully automate in-vitro fertilization
- I enjoyed using C++ to develop solutions and help parents create families
- Whereas previously I released software features to assist fertility doctors, I am now looking to continue developing software tools and features for the File System Service team

## About Oracle

- Oracle is a software company that functions primarily in enterprise software and cloud computing
  - *Enterprise Software*: Oracle sells software products (specifically to businesses) such as Oracle Database and Enterprise Resource Planning (ERP)
    - ERP is a software tool used by businesses to help manage finances, human resources, and supply chain management
  - *Cloud Computing*: similarly to Amazon Web Services, Oracle provides its own cloud computing service OCI so that clients can purchase computing resources on a per-use basis

## About Oma Robotics

- Medical manufacturing company committed to improving IVF success rates using AI and robotics
- Worked primarily on developing algorithms and user interfaces for the Sperm Select App (SSA), a software that looks at a sperm cell and tells fertility doctors if it is healthy or unhealthy using AI
- SSA ran on a dedicated Linux PC, these PC's were deployed nationwide and used within fertility clinics

## Sperm Healthiness by Morphology

- *Situation:* At the time, fertility doctors using SSA were only able to see if a sperm cell itself is overall healthy or unhealthy
  - Though, fertility doctors were not able to understand *why it was healthy*
  - Although a sperm cell can look healthy, it can still have a strange looking tail (for example)
  - AI Engineers had a new AI ready that could classify healthiness by sperm morphology (i.e. sperm head, neck, and tail)
- *Task:* to work alongside AI Engineers and develop a new user interface to *show healthiness by morphology* (i.e. show that the head is healthy, tail is unhealthy, neck is healthy) within SSA
- *Action:*
  - I realized that a majority of my work was very closely related to other previously completed work (i.e. how overall sperm healthiness is displayed within SSA)
  - I also realized that SSA had very little room to add lots of buttons and screens, so I had to *design a user interface feature that was able to show lots of data within limited space*
  - I dug through the codebase, and referred to similar work in order to streamline completion of this task
  - I also drafted up a space efficient user interface that used toggleable buttons in order to switch between seeing healthiness for head, neck, and tail
- *Results:* developed a UI feature where fertility doctors can see two new things:
  - 1) Individual healthy / unhealthy classification by head, neck, or tail (i.e. head is healthy, but tail is not)
  - 2) Histogram data of how the healthy / unhealthy classification is made
  - Fertility doctors would now able to see *what* makes a sperm cell healthy or unhealthy

## Pipette Calibration Algorithm

- *Situation:* other engineers were developing robotic algorithms that required a robot moving a pipette
  - A pipette is needed when working with oocytes (i.e. unfertilized female eggs)
  - The robot is not able to accurately move the pipette to some position (unless some calibration is performed)
- *Task:* to develop an algorithm that calibrates the pipette so that pipette moves are as accurate as possible
- *Action:*
  - I sat down with our Robotics and AI Engineers to scope out the necessary steps that needed to be performed within this algorithm
  - I also realized that a lot of the work that needed to be complete was very similar to another algorithm (i.e. the needle calibration algorithm)
  - I dug through the codebase and often referred to similar algorithms in order to streamline completion of this algorithm
  - I maintained a close line of communication with our Robotics Engineer in order to ask for support when needed
- *Result:*
  - Algorithms were able to move pipettes to any position with high accuracy
  - This algorithm ran prior to investors / stakeholders viewing our recent major development
  - This pipette calibration algorithm can be initiated at any time with a simple button click within the software

## Needle Injection Algorithm

- *Situation:* a new algorithm needed to be developed in order to have a robot emulate a fertility doctor fertilizing a female egg
- *Task:* to develop an algorithm where a robot fertilizes a female egg
- *Action:*
  - I sat down with our Robotics Engineer to scope out the steps needed within this algorithm
  - Mid-development, I found that the biggest issue was whether the needle would successfully puncture through the female egg (i.e. the outer layer of the female egg can sometimes be difficult to puncture through)
  - After communicating with embryologists, we devised the needle puncture would be one of two cases:
    - 1) The needle punctures through the female egg on the first try, and the algorithm can continue as needed
    - 2) The needle puncture fails on the first try. The algorithm will detect this failed puncture, and perform a rapid in-out motion with the needle to “slice” through the female egg
- *Results:*
  - A robot was now able to successfully emulate the same fertilization procedure that fertility doctors perform and ultimately fertilize a female egg
    - This algorithm was often featured to investors / current stakeholders to highlight major R&D developments
  - I implemented two failsafe measures:
    - 1) Algorithm ensures that the needle does not puncture too far past the female egg
    - 2) Algorithm ensures the female egg is oriented in such a way that dangerous regions are not stabbed (or else the female egg would die immediately)
  - This needle injection algorithm can be initiated at any time with a simple button click within the software

## Stepper Motor API

- *Situation:* a stepper motor is needed in order to extract sperm cells from a petri dish. The current setup took up a lot of space, and was nowhere near release ready
  - Our Robotics Engineer proposed purchasing an Arcus Stepper Motor in order to provide a more space efficient solution
  - This new stepper motor required developing an entirely new library to control this stepper motor
  - This API is needed so other robotic algorithms can successfully extract sperm cells from a dish
- *Task:* to develop a C++ library to control the new Arcus Stepper Motor
- *Action:*
  - Arcus provided a Python GUI interface to control the stepper motor
  - I dug within the provided Python code, and found the actual messages that were being sent to the stepper motor in order to control it
  - Ultimately by referring to similarly completed work, I found that developing a C++ library simply required setting up a communication channel from a PC to the stepper motor, and sending those same messages to the stepper motor
  - Testing the C++ library was straightforward: simply call the C++ functions (from the library) and see if the stepper motor performs as expected
- *Result:*
  - A new C++ library was developed to control the Arcus Stepper Motor
  - The library is capable of:
    - Setting / reading the current velocity, acceleration, and jerk
    - Having the stepper motor spin a finite amount of steps
    - Having the stepper motor spin infinitely
  - Was able to have a new space efficient setup that was release ready, and have finer control of extracting sperm cells from a petri dish

## New Relic Monitoring Dashboard

- *Situation:* the SSA software would sometimes run into **fatal bugs**. These bugs impeded fertility doctors from using the software
  - Diagnosing the bugs was **tedious process**
  - Required remotely logging into the computer (that the software ran on) and sifting through huge log files
  - Engineers proposed using New Relic, a real-time monitoring platform that tracks and provides information about software
- *Task:* to develop a real-time monitoring platform that **centralizes all important information so that bugs can be detected and solved more quickly**
- *Action:*
  - After reading documentation, I set up the New Relic Infrastructure Agent on all nine PC's that used SSA
    - The Infrastructure Agent is needed so that New Relic can send data from the PC's to the New Relic platform
  - Data that was being sent (from the PC's) included:
    - Computer metrics (i.e. GPU usage, CPU usage, SSD space)
    - Log files
    - SSA software version numbers
  - In order to develop dashboards with the new data, I had to learn a SQL-like language
- *Results:*
  - Using the SQL-like language I had just learned, I developed several new charts and dashboards that displayed:
    - Line graphs of GPU usage per each PC
      - So that we know the AI is working when SSA is running
    - Percentage of available and consumed SSD storage capacity
      - So we can (1) ensure any video recordings taken are uploaded to the cloud and (2) ensure fertility doctors are able to log into the PC
    - Version numbers of SSA on each of the nine PC's
      - To ensure that the software updated properly
    - Number of fatal errors and minor bugs
    - Log files (filterable by either errors or warnings)
  - In order to stay ahead of issues, I also **integrated an alert notification system**
    - When a fatal issue will occur soon (i.e. space consumed by SSD is nearing 90%), a Slack message warning is sent
  - Sifting through log files required accessing the New Relic platform (instead of remotely logging on and locating the log file on each of the nine PC's)
  - Issues (that were previously detected too late) were now able to be detected ahead of time due to the alert notification system

## Automated Data Collection Pipeline

- *Situation:* when fertility doctors use SSA, they take many screen recordings that are gigabytes in size
  - If the PC (that fertility doctors are using) runs out of SSD space because they took a lot of screen recordings, fertility doctors won't be able to use the software
  - AI Engineers also need these screen recordings so that they can train up the AI. They typically request to access these videos, and retrieving these videos is a very tedious full-day process
- *Task:* to figure out a way to efficiently deliver these screen recordings (from the PC's) and to the AI Engineers so that fertility doctors can use SSA without any issues
- *Action:*
  - I devised two scripts to automate everything:
    - 1) Upload program: take video recordings (from the PC's) and upload them to a cloud database
    - 2) Download program: take the video recordings (now on the cloud database), and download them so that AI Engineers can access them
  - I opted for using the appropriate technology for the task – in this case, Python (instead of shell scripting)
    - Python was a language that I was more versed in (so it would be more time efficient to create these scripts)
    - AWS also provided easy-to-read documentation on how to upload / download data from the cloud using Python
  - I familiarized myself with Amazon S3 – the AWS cloud database service that we were going to upload the video recordings to
  - To ensure that these programs would run every day, I automated these scripts within crontabs – a Linux utility tool that runs programs on a schedule
- *Results:*
  - Both the upload and download programs were scheduled to run every time of the day, and ultimately resolved the issue with screen recordings
  - Because the amount of available SSD space was mostly available, fertility doctors are able to now log into the PC's with no issues
  - Because the screen recordings (taken by fertility doctors) are now readily available, AI Engineers are able to more quickly train up the AI

## (Oracle) Questions Possible Asked

Why are you interested

- I am interested because previously I enjoyed writing code to create impact at Oma Robotics, and I am now looking to continue writing code to create impact within the OCI team
  - At Oma Robotics: I worked on a software that used AI to identify how healthy or unhealthy a sperm cell is
  - This software was used by fertility doctors during real medical procedures
  - I enjoyed knowing that the software features I was developing was high impact
    - In the sense that, my code was helping parents create families
  - Joining the OCI team gives me that opportunity to continue writing code to create impact
  - OCI is a popular cloud service used by many businesses, and a lot of the code I would be writing would be directly affecting them
    - Plus, the job listing mentions that engineers will have a “significant impact on the... business aspects of the project”
  - Ultimately, I am interested in the position because I see it as an opportunity to continue my ambition of writing code to create impact

What are your career goals

- I am looking to deepen my C++ knowledge within the next two to three years
- That entails two things: collaborating with Senior Engineers to learn from them, and completing as much big ticket projects within the team
- Having the opportunity to realize that career goal is definitely possible within the OCI team
  - The OCI team uses C++ (and other related tools like GDB) to maintain and develop features the File Storage tool, so reaching a high-detail understanding of C++ is realistic
  - The position also mentions that I get the opportunity to work at any part of the stack (including very low-level), so that gives me the opportunity to really prove my C++ technical expertise in very complex projects
  - The OCI team clearly has Senior Engineers with amazing talent, and getting the opportunity to bounce ideas off of them and learn from them is also realistic
- I believe that making that career goal a reality is indeed possible within the OCI team



## (J&J) Questions Possible Asked

Why are you interested?

- I am interested in this position because I see that my skills can help solve problems within the team
- The job listing outlines that the team is looking for:
  - “Medical device development experience”
    - Oma Robotics was also a medical device startup aimed at improving fertility success rates
    - I was worked on a software called the Sperm Select App (SSA), which was used by fertility doctors and able to look at sperm cells and identify it as healthy or unhealthy
    - I frequently developed UI features and robotic algorithms for SSA
      - UI features: auto-stop video recordings, sperm healthiness by head / neck / tail morphology
    - I have experience validating medical device software
      - For example: I frequently worked with unfertilized female eggs to check if a robotic algorithm was able to successfully puncture a needle through the egg
  - “Solid understanding of GUI development in C++, Qt, Git, and Linux environments”
    - C++, Qt, and Linux environments are all tools I primarily used at Oma Robotics
    - SSA ran on a Linux PC
      - Tasks like remote logging in to other Linux PC’s were all too familiar tasks
    - I used C++ to develop...
      - A library to control a stepper motor in order to extra sperm cells from a petri dish
      - We also used Git for standard version control
    - I also used Qt to develop user interface features for our SSA
      - 1) Show *why* a sperm cell is healthy using AI
- I’m looking to apply my current skills and knowledge in order to help problem solve
- I’m also looking to complete challenging tasks within the J&J team, and continue building my current technical expertise to help deliver a software that makes impact within digital surgery

How have your past experiences influenced your decisions for your career?

- My software engineering experience at Oma Robotics made me realize I enjoy working where my code creates impact
- At Oma Robotics, I frequently developed and tested the Sperm Select App (SSA)
  - SSA is a software used by fertility doctors
  - SSA looks at a sperm cell and determines if it is healthy or not
- Previously, I was tasked to install SSA within a fertility clinic located within Novato, California
- Set up involved setting up the Linux PC, installing the software (within the PC), and testing the software with live sperm samples (to ensure the AI runs properly)
- After completing the set up, fertility doctors were astounded by how accurate the AI was in analyzing sperm cell healthiness, and how useful of a tool it became during IVF procedures
- I realized that working within a MedTech company meant that the code I was writing will create impact – in this case, impact within parents struggling to create families
- I'm seeking to remain in the MedTech sector where the code I write creates impactful results
  - That is also why I am primarily interested in the GUI Software Engineering position at J&J
- J&J is indeed involved in the MedTech sector to tackle complex issues such as orthopedics and vision using technology
- In this case specifically: I am looking to continue writing easy-to-use frontend software features for surgeons, and create solution plans for surgery patients

Tell me about your job history

- I previously worked within a small organization called Pocket Racers within University of California, Los Angeles (UCLA)
- Pocket Racers aimed at promoting fully autonomous RC cars as easy-to-build and affordable
- I worked on projects such as developing a fully responsive website (from scratch) and migrating the Robot Operating System of an RC car from Python3 to C++
  - As a result of the ROS project: I found that I enjoyed working on complex tasks involving data structures and algorithms
- I would then transition to becoming a Computer Science Instructor & Hiring Manager at Juni Learning
- I would continue my interest in working with complex tasks by breaking down complex programming topics to students as young as middle school
- I would then finally transition into professional Software Engineering within Oma Robotics
- I would still continue my interest with complex tasks by...
  - 1) Writing two robotic algorithms that mimic the same IVF procedures that fertility doctors perform
- I also realized that remaining within the MedTech sector was the best opportunity to continue my interest in tackling complex tasks
- I really appreciated how the code I was writing to solve challenging projects was making an impact
  - In the case of Oma Robotics: the code I was writing was almost directly helping create families nationwide
- I'm looking to continue remaining in the MedTech sector at J&J to continue writing code for challenging projects, and make impactful results for patients

What are your career goals?

- Having the opportunity to develop software at J&J aligns with my career goals
- My overall career goal can be divided into short term and long term goals:
  - **Short term:** I would like to continue working with C++ through my career (mostly because of its versatility ranging from GUI development to GPU Programming)
  - Gain and refine my C++ expertise by working within the same project for a long period of time
    - I got experience doing just that at Oma Robotics by working on the Sperm Select App (SSA)
    - Within SSA: I developed GUI features, robotic algorithms, and C++ libraries that interfaced with stepper motors
    - At J&J: I'm looking to continue that by continuing to develop GUI features for a software controlling surgical robots
    - That would mean completing as much big ticket items to refine my knowledge
  - **Long term:** Be part of a team that develops and delivers a product from scratch
  - Primarily involves leveraging the C++ technical expertise (that I would have previously gained from my short term goal realization) to deliver a software that provides a use case for an unmet need to clients
  - At a MedTech company like J&J: creating impact is definitely possible since the code I would be writing directly affects patients
- I believe that making that career goal outline a reality is indeed possible at J&J

## Strengths and weaknesses?

- Strength: communication skills
  - Working within a MedTech company made me realize how important communication is
  - Obviously, I have limited understanding of fertility as a Software Engineer
  - At Oma Robotics: I frequently had to interface with embryologists in order to get a high level understanding of what embryology is
  - I would then use that understanding to delicately approach an algorithm that moves an unfertilized female egg
  - Ultimately: my communication skills is a strength because it enables me to more easily collaborate with my colleagues in order to scope out complex projects
- Weakness: the urge to know everything
  - Because technology is changing everyday, I feel the need to stay up to date with that pace by learning something new everyday
  - But as a result: that also means I have some knowledge of a lot of skills, but less mastery of a honed set of skills
  - I realized that I should first approach tasks by seeing if it can be done with the technical expertise that either I or the team knows
  - If it does: try to work with that technology and create a nice solution
  - If it doesn't: use my communication skill strength to converse with my colleagues and see if a new tool / framework is *necessary*
  - Ultimately: my urge to know as many tech tools as possible functions as a two way street:
    - I gain some expertise in a lot of different tools and programming languages
    - That also means I lack honed knowledge in a finite set of skills
    - I counteract that by leveraging my communication strength to collaborate with my colleagues and generate solutions using the technical expertise that we currently know

## (Ocado Group) Questions Possible Asked

Why are you interested?

- I am interested in this position because I see that my skills can help solve problems within the team
- The job listing outlines that the team is looking for someone with:
  - “Competency in... C++ and Python, and the Linux Operating System”
    - C++ and Python3 are programming languages I developed with on a Linux PC at Oma Robotics
    - I used C++ to develop robotic algorithms that mimic the same exact IVF procedures that fertility doctors perform today
    - I used Python3 and AWS S3 to automate our data collection pipeline, thereby significantly accelerating the training process for our AI
  - “Excitement at the prospect of getting your hands dirty with a variety of technologies and tools”
  - Oma Robotics was indeed a startup, so learning several new tools and platforms in a fast environment are things I am all too familiar with
  - For example: I previously had to learn a new database language in order to develop analytical dashboards (within the New Relic platform) in order to diagnose software bugs in real-time
  - The idea of quickly learning new tools in order to solve challenging tasks is something I have experience in
- I’m looking to apply my current skills and knowledge in order to problem solve within the team
- I’m also looking to take on challenging projects within the team to help deliver software solution that makes impact within the grocery distribution sector

Describe your background and how it matches with the job

- I began my career as a Software Engineer at Oma Robotics
- Day-to-day tools include:
  - C++ for software development
  - Python3 and Bash to automate redundant tasks
  - Git for standard version control
- Primary responsibilities:
  - Creating and distributing new release builds of our Oma Robotics software
  - Implementing C++ robotic algorithms to mimic the same fertilization process performed by fertility doctors today
  - Developing new UI features as requested by our software users (i.e. fertility doctors)
- A lot of what the job listing mentions is what I have experienced in as a result of Oma Robotics
  - “Competency in programming languages such as C++, Python, and Linux”
  - We used C++ to develop our robotic algorithms and interface with hardware at a low-level
  - We used Python3 to provide quicker turnaround for retraining our AI by fully automating our data collection pipeline
  - We developed and debugged the Oma Robotics software on LinuxOS
  - “Background in robotic manipulation... and computer vision”
  - **Robotic Manipulation:** I used C++ API's to control robots holding needles and pipettes
  - **Computer Vision:** I used CV in order to get the  $(x,y)$  pixel position of an unfertilized egg (for example)
    - Getting the  $(x,y)$  pixel position was necessary to ensure that (for example) the needle does not inject through an egg (thereby killing it)
    - Technically speaking: the Oma Robotics software collected images from a microscope every  $t$  frames, and these images were preprocessed by an AI through an image acquisition pipeline
    - Every  $t$  frames, the AI updates the  $(x,y)$  position of an unfertilized egg (for example) object
    - Every robotic algorithm that wants to get the  $(x,y)$  position of that unfertilized egg requires maintaining a data member of that object
- Ultimately: a lot of the experiences I gained through Oma Robotics does indeed align with what your company is searching for in a potential candidate
- I'm looking to leverage the same technical expertise and experiences I developed at Oma Robotics to problem solve within the grocery distribution sector

What are your expectations from the role

- My understanding of the role's expectation is that the team is looking for a Software Engineer who can get their hands dirty in different technical domains and produce significant results
- I believe I can meet that understanding
- Having worked at Oma Robotics, I also did also produce results in varying domains such as UI development, robotic algorithm development, and software analysis tools
  - **UI Development:** Using C++, I developed a new GUI interface so that fertility doctors can understand *why* a candidate sperm cell is healthy
  - **Software Analytics:** Using a SQL-like language, I developed a real-time analytical dashboard to quickly bugs from the same software deployed in clinics nationwide
- My other expectation of the role is personal growth
  - Growth to me is something I do value
  - Considering that the position listing mentions "Excitement in getting your hands dirty with a variety of tools and technologies," I anticipate that personal growth is realistic to expect
  - Going about realizing that expectation involves completing as many big ticket projects as possible to gain as much technical experience
- Ultimately: I expect to write code and produce results in various technical domains
- Luckily, that also goes hand-in-hand with my expectations of personal growth



Job history / about yourself?

- I previously worked with a small organization called Pocket Racers within University of California, Los Angeles (UCLA)
- Pocket Racers aimed at promoting fully autonomous RC cars as easy-to-build and affordable
- I worked on projects such as developing a fully responsive website from scratch and migrating the Robot Operating System of an RC car from Python3 to C++
  - As a result of that project: I realized I enjoyed working on complex tasks involving data structures and algorithms
- I would then transition to becoming a Computer Science Instructor & Hiring Manager at Juni Learning
- I would continue my interest in working with complex tasks by breaking down complex programming topics to students as young as middle school
- I would then finally transition into professional Software Engineering within Oma Robotics
- I would still continue my interest with complex tasks by...
  - 1) Writing two robotic algorithms that mimic the same IVF procedures that fertility doctors perform
- I also realized that remaining within the MedTech sector was the best opportunity to continue my interest in tackling complex tasks
- I really appreciated how the code I was writing to solve challenging projects was making an impact
  - In the case of Oma Robotics: the code I was writing was almost directly helping create families nationwide

I'm looking to continue remaining in the MedTech sector at J&J to continue writing code for challenging projects, and make impactful results for patients

What are your career goals?

- I would like to grow my technical expertise as a C++ Software Engineer
- As a rough outline of how I would achieve that:
  - 1) Gain and refine my C++ expertise by working within the same project for a long period of time
    - Given the opportunity to be a C++ Engineer at the Ocado Group, that would mean completing as much big ticket items to refine my knowledge (within a two to three year timeframe)
    - May include modifying build systems or developing new robotic algorithms (to name a few)
  - 2) Reach Mid-level / Senior Engineer levels by climbing vertically within the company
    - Would entail branching out to other Engineering positions internally within the team
    - With my technical expertise in C++, those other positions would be Machine Learning Engineer (where I would be GPU / CUDA programming with C++ for example)
  - 3) Be part of a team that develops and delivers a product from scratch
    - Would be a true testament of applying the C++ expertise I have learned in the past

I believe that making that career goal outline a reality is indeed possible at the Ocado Group

## Strengths and weaknesses?

- Strength: communication skills
  - Previously working within a MedTech company made me realize how important communication is
  - Obviously, I have limited medical expertise as a Software Engineer
  - At Oma Robotics: I frequently had to interface with embryologists in order to get a high level understanding of what embryology is
  - I would then use that understanding to delicately approach an algorithm that moves an unfertilized female egg
  - Ultimately: my communication skills is a strength because it enables me to more easily collaborate with my colleagues in order to scope out complex projects
- Weakness: the urge to know everything
  - Because technology is changing everyday, I feel the need to stay up to date with that pace by learning something new everyday
  - That may include staying learning a new JavaScript frontend library to stay up to date with popular technology trends
  - But as a result: that also means I have proficient knowledge in a lot of skills, but less mastery of a honed set of skills
  - I realized that I should first approach tasks by seeing if it can be done with the technical expertise that either I or the team knows
  - If it does: try to work with that technology and create a nice solution
  - If it doesn't: use my communication skill strength to converse with my colleagues and see if a new tool / framework is *necessary*
  - Ultimately: my urge to know as many tech tools as possible functions as a two way street:
    - I gain some expertise in a lot of different tools and programming languages
    - That also means I lack honed knowledge in a finite set of skills
  - I counteract that by leveraging my communication strength to collaborate with my colleagues and create solutions using the technical expertise that we currently know

What was a challenge you experienced, and how did you overcome it?

- *Situation:* I was tasked with using the New Relic platform to create a real time monitoring dashboard and centralize important data about SSA (the software used by fertility clinics)
  - The goal of this task was to detect fatal bugs before they would happen
  - Creating dashboards on the New Relic platform required learning a SQL-like database language (of which I had little to no experience in)
- *Task:*
  - The challenge is to quickly learn a SQL-like database language in order to create dashboards that detect fatal bugs ahead of time
- *Action (i.e. how I overcame it):*
  - **Research:** I essentially read every available documentation (provided by New Relic) in order to learn database concepts and fundamentals
  - **Experiment:** every time I read about a new database concept, I would write hundreds of database statements (within a playground) to see how the concept worked in actuality
    - If a database statement was incorrect, I delved into it and deeply understood *why* it was incorrect
  - **Support:** I outreached to my colleagues to see if they were also familiar with this SQL-like language
    - Unfortunately, they too were not familiar – but pointed out that the language was very close to SQL (another database language)
    - I then caught up with SQL documentation
    - Cross referencing with another similar database language gave me more resources to refer to
- *Results:*
  - Using the SQL-like language that I had learned, I created several new dashboards that centralized important data such as:
    - Computer metrics (i.e. GPU usage, amount of available space on SSD, etc)
    - Number of current fatal errors and minor bugs
    - Software version numbers
  - I also went **above and beyond** by integrating an alert notification system to stay ahead of fatal issues
    - When a fatal issue will occur soon (i.e. space consumed by SSD is nearing 90%), a Slack message warning is sent to the Engineering Team so that it can be addressed immediately
  - I realized that learning new tools for an unfamiliar task requires a dedicated amount of research and experimentation

- I also realized that asking for support can be helpful in order to reduce steep learning curves for difficult

The next time I need to learn a new tool for a given task, I would continue the same methodology: extensive research, experimentation, and asking for support when needed

## (TabaPay) Questions Possible Asked

Why are you interested?

- I am interested in the position because I enjoy writing code when it creates impact
  - Anecdotally speaking, that was a **primary reason** I worked as a Software Engineer at Oma Robotics
  - Oma Robotics was a Medical Tech company that increased fertility success rates using robotics and artificial intelligence
  - A lot of the code I was writing was directly helping families
    - For example: in order to provide quicker turnaround for training our AI, I used Python3 to automate the transfer of video data (from the clinics to our Engineering team)
  - Whereas previously I created impact within the MedTech sector, I'm looking to **continue creating impact**
  - I'm looking to continue creating impact within the FinTech sector by writing code to **deconstruct the myth of instant payment systems being costly and risky**
  - I'm looking to use Python3 to implement new API calls for the TabaPay Unified API so that clients (of the API) can provide **instant payrolls** to their employees (for example)
  - Ultimately: I am interested in the Software Engineering position at TabaPay because I see it as a way to continue writing code to create impact – particularly within the FinTech sector

What are your career goals?

- I'm looking to focus on personal growth by tackling difficult projects and solve challenging tasks
- In my early career: I aim at becoming proficient in backend languages such as Python3 and Golang
  - I'm looking to pick up big ticket tasks to improve my technical expertise
  - Those big ticket tasks might be identifying performance bottlenecks for the Unified API
  - Or researching a new security protocol to ensure safe transfer of money between two systems
- In my Mid career: I aim to be a leading developer within the team
  - I plan on leveraging the Python3 and Golang expertise I would gain from my early career, and use it provide technical consulting for a product
  - I aim on collaborating with other Mid-level Engineers to construct stories for the next Agile Sprint (for example)
  - I aim on maintaining direction for a product (whether it's the Unified API or a new Tabapay software) by also writing efficient code solutions
- Ultimately: I aim to learn as much as I can by completing as many challenging tasks as possible
  - Considering Tabapay is indeed a startup environment where wearing many hats is required, I do indeed believe that realizing that personal growth ambition is realistic

## (Lucid Motors) Questions Possibly Asked

Why are you interested?

- I am interested in this position because I see that my skills can help solve problems within the team
- The job listing outlines that:
  - “This position requires experience working with the user interfaces development”
  - I have two years of UI development experience, both in C++ and in JavaScript
  - C++: Using the Qt framework, I developed a UI feature that enabled fertility doctors to understand why a sperm cell is healthy or unhealthy
  - JavaScript: Using React and JavaScript component libraries, I developed several responsive websites
  - These websites featured content such as my professional work experience, and how to build your own AI self-driving RC car from scratch
  
  - “Our ideal candidate possesses the ability to learn quickly... in a fast-paced environment”
  - I have startup experience, where learning many new tools and technologies is required in a fast-moving environment
  - I also had limited robotics knowledge coming into Oma Robotics
  - But by the end of it: I worked on and developed two robotic algorithms (i.e. needle injection and pipette calibration algorithms)
  - I can articulate on these experiences later if you would like me to
  
  - “Expertise in C++... with high proficiency to write clean and well-structured code”
  - I understand C++ both at a low and high level
    - Low level: previously wrote C++ libraries that establish communication channels to hardware, and send literal raw bits of data in order to control motors
      - Debugged complex multithreaded software at Oma Robotics
    - High level: previously developed many UI features within SSA
      - I also understand concepts from advanced data structures to multithreading
- I not only see an opportunity to problem solve, but also I see the position as an opportunity for personal growth
- I would love to:
  - Refine my UI design skills by developing infotainment features using the Qt framework



- Learn new technical skills and frameworks (in a fast-paced environment) to solve big ticket issue
- Ultimately, I am interested in the C++ Engineer position because I see it as an opportunity as both a problem solving and personal growth opportunity

C++ and Qt understanding level?

- Very familiar with C++
  - A language I commonly taught at Juni Learning as a Computer Science Instructor
  - Required breaking down difficult concepts (such as pointers and classes) to students as young as twelve years old
  - Also familiar with popular C++ libraries such as **Boost and Eigen** (to perform Linear Algebra calculations)
- Very familiar with Qt fundamentals and concepts
  - **Slots and signals**: enables widgets to send signals containing event information which can be received by other widgets
  - **QML**: markup language (similar to CSS) to style widgets
  - **QMake**: a high-level language used to automate Makefile generation
    - Makefiles are needed to convert source code to binary executables
- A majority of my Qt expertise comes from developing the SSA software at Oma Robotics
  - I have also previously used Qt in undergrad
  - As part of my capstone projects for the course, my team developed an overhead shooter arcade game
  - Essentially attack as many computer opponents within a time limit, highest score wins

What was a challenge you experienced, and how did you overcome it?

- *Situation:* I was tasked with using the New Relic platform to create a real time monitoring dashboard and centralize important data about SSA (the software used by fertility clinics)
  - The goal of this task was to detect fatal bugs before they would happen
  - Creating dashboards on the New Relic platform required learning a SQL-like database language (of which I had little to no experience in)
- *Task:*
  - The challenge is to quickly learn a SQL-like database language in order to create dashboards that detect fatal bugs ahead of time
- *Action (i.e. how I overcame it):*
  - **Research:** I essentially read every available documentation (provided by New Relic) in order to learn database concepts and fundamentals
  - **Experiment:** every time I read about a new database concept, I would write hundreds of database statements (within a playground) to see how the concept worked in actuality
    - If a database statement was incorrect, I delved into it and deeply understood *why* it was incorrect
  - **Support:** I outreached to my colleagues to see if they were also familiar with this SQL-like language
    - Unfortunately, they too were not familiar – but pointed out that the language was very close to SQL (another database language)
    - I then caught up with SQL documentation
    - Cross referencing with another similar database language gave me more resources to refer to
- *Results:*
  - Using the SQL-like language that I had learned, I created several new dashboards that centralized important data such as:
    - Computer metrics (i.e. GPU usage, amount of available space on SSD, etc)
    - Number of current fatal errors and minor bugs
    - Software version numbers
  - I also went **above and beyond** by integrating an alert notification system to stay ahead of fatal issues
    - When a fatal issue will occur soon (i.e. space consumed by SSD is nearing 90%), a Slack message warning is sent to the Engineering Team so that it can be addressed immediately
  - I realized that learning new tools for an unfamiliar task requires a dedicated amount of research and experimentation

- I also realized that asking for support can be helpful in order to reduce steep learning curves for difficult
- The next time I need to learn a new tool for a given task, I would continue the same methodology: extensive research, experimentation, and asking for support when needed

How do you handle pressure?

- I handle pressure by assigning setting priority levels to tasks I have to complete
- At Oma Robotics, I was assigned three tasks to complete concurrently:
  - 1) Pipette Calibration Algorithm
  - 2) Needle Injection Algorithm
  - 3) Real Time monitoring dashboard to diagnose fatal bugs
- I found that completing the Pipette Calibration Algorithm had the highest priority
  - Other members of my team were working on other algorithms that required working with pipettes
  - Naturally: this task gets prioritized first (since other people's work is now dependent on my work being completed)
- The Needle Injection Algorithm simply required completing relatively simple tasks
  - For example: have the robot stop moving a needle past some point or else it will kill the female egg
  - So essentially when I was taking a brief break from completing the Pipette Calibration Algorithm, I would switch gears to completing the easier tasks for the Needle Injection Algorithm
  - Essentially maximizing time efficiency by completing less complex tasks during downtimes
- The Real Time Monitoring Dashboard took the least precedence
  - My supervisor mentioned expressed that it was a low priority ticket item
- Ultimate takeaways:
  - I was able to complete the three tasks by assigning appropriate priorities
  - Triage tickets to assess priority levels
  - Identify whether my work is dependent on others, and assign that the highest priority
  - Maximize my time efficiency by using break periods to complete less complex tasks

What are your career goals?

- I would like to grow my technical expertise as a C++ Software Engineer
- As a rough outline of how I would achieve that:
  - 1) Gain and refine my C++ expertise by working on the same team for a long period of time
    - Given the opportunity to be a C++ Engineer at Lucid Motors, that would entail completing as much big tickets to refine my knowledge in a two to three year timeframe
    - May include modifying build systems, developing new UI features (to name a few) for the infotainment system, or creating a new SDK from scratch
  - 2) Reach Mid-level / Senior Engineer levels by climbing vertically within the company
    - Based on other job listing, C++ looks to be the primary language of choice – so staying within the company gives me an opportunity to really gain a deep understanding of C++ and other relevant frameworks
    - That would also entail branching out internally to other teams – such as the Autonomous Driving teams
  - 3) Be part of a team that develops and delivers a product from scratch
    - Would be a true testament of applying the C++ expertise I have learned in the past
- I believe that making that career goal outline a reality is indeed possible at Lucid Motors

## Questions to Ask

- What opportunities are there for professional development at Oracle?
- What are the next steps in this interview process?