COMP3004 – Deliverable 1 – Proposal

tutor|onight

What is our project?

For our project we decided to build an iOS mobile application with the goal of matching students who need help with their classes with tutors who have validated their skills – in a mobile, on-demand, cashless environment.

Why is it Interesting?

We wanted to build something that we envision could be widely used, scalable, and serves a purpose. Our application would allow students to log into their account and request a tutor for a specific course or subject for a given timeframe, which would then notify tutors who match that request (in terms of skill and availability) and they can choose whether to accept or decline the session.

Our Justification

Tutors today advertise themselves all over Carleton by posting papers everywhere. We feel this is out-dated and wanted to modernize the process. We spent a lot of time praising the successful "uber" business model where they are essentially the middle-man, but provide a way to track clients/drivers with a profile – while providing a safe, cashless environment. That model now has expanded to similar apps/markets including: food delivery (uberEats , skipTheDishes), on-demand snow plow/lawn care (plowsAndMowz, touchPlow). Some common/key-points include:

- *on-demand* - you don't need to commit to anything (like winter plow contracts) and just use it as needed
- *cashless* – safer to pay in electronic form and not have to worry about cash
- *trusted* – drivers/businesses have profiles where clients can rate the service and provide feedback

We would like to implement all these concepts into our app:

- on-demand - need a tutor tonight for tomorrows big test?
- cashless – tutor never worries about getting cash payment, secure with apple pay
- trusted – tutor must submit their transcript to verify they successfully (A-) completed the courses they claim to have tutoring abilities, students can rate the tutors

Why does it make sense in a mobile factor?

- Not everyone carries a computer around, but everyone has a smartphone these days
- Apple has recently opened up their apple-pay API for developers – perfect for a secure cash-less environment as well as location based services for tutor/student tracking information

dDevelopers
Satinder Singh – 100742575 / Atamjeet Singh – 100842821
Corey Cougle – 100745291 / Aarzan Irani - 100826631

1. **Functional Requirements**:

    1.1. Account Creation Interface.
        1.1.a. Creates a required Student account.
        1.1.b. Allows for the creation of a Tutor account with higher requirements.
    1.2. Sign-in and Account Verification Interface.
    1.3. Apple pay third party app registration.
    1.4. General Navigation Interface.
        1.4.a. Displays code options (COMP) and year level (3000).
    1.5. The application must access a database of Courses and algorithmically selected Tutors.
    1.6. Provide the ability to communicate for the purpose of coordinating a Tutorial Session location.
    1.7. Integration of Apple Pay for the transferal of payment.
    1.8. Two-way rating system for upkeep of user integrity.

2. **Non-Functional Requirements**:
    2.1. A Student should never have to wait more than 1 second for an action to be performed, except when waiting on a response from Tutors.
    2.2. A potential Tutor should be appraised for suitability by OCR.
    2.3. UI should be so simple it appears familiar at first sight.
    2.4. App crashes should never happen.
    2.5. Light weight.
    2.6. Specific content should be notable by the Student during a request.

User Scenarios:

A) A struggling Student is in need of a provably intelligent Tutor to aid in understanding a key concept from class, either for a test or an assignment. The Student has limited and unreliable resources for finding a suitable Tutor. These would be flyers containing tear-off phone numbers, or exorbitant Tutor services with non-ideal scheduling. The Student needs a **Tutor Tonight**. The Student downloads this app and creates an account by registering their School email address. An email address confirmation is made and the Student is presented with the tutorTonight Home screen. The Student uses the app to request a Tutorial Session at 6:30pm for COMP3804 at the Library. The Student sees a price per hour value associated with the course level and a price of the entire Session given the duration requested. The Student agrees to the pricing and submits the request. Once a Tutor agrees to the Tutorial Session, the Student is able to textually (in app) coordinate location specifics with the Tutor. The duo meets for the Session and the Student confirms the initiation of the Session. Once the Session has been successfully ended, payment is confirmed and processed. Immediately, the Student is presented with a Rating opportunity to verify and upkeep the integrity of the unique Tutor's abilities. The Student is satisfied and provides and 5-star rating with a positive comment on the level of knowledge and concise communication skills of the Tutor.

dDevelopers
Satinder Singh – 100742575 / Atamjeet Singh – 100842821
Corey Cougle – 100745291 / Aarzan Irani - 100826631

B) An academically successful Student is in need of coffee funds, but doesn't want to commit to a part time job due to time constraints involved with higher education. The Student has already verified themselves as a Tutor on tutorTonight. They receive a push notification from their tutorTonight app notifying them of an available Session. The Session is requested for 6:30pm at the Library for COMP3804. The Session fits the Tutor's schedule and preferred courses, so they accept the Session. They are now able to communicate with the Student in order to coordinate a more specific location. The duo meets up and the Tutor confirms the initiation of the Session. They share a successful Session. Once the Session has been ended, the Tutor can indicate to tutorTonight that the Session is complete with the press of a button. Immediately after the Session has been confirmed ended, the Tutor is presented with a Rating opportunity to maintain a mutually beneficial Tutor-Student environment. The Tutor experienced no issues with the Student, and therefore leaves an A+ rating. The Tutor can gladly go get some coffee.

Non-Functional Justifications

1. The application needs to be almost instantaneous while browsing from each tab. If the application doesn't run smoothly, users will not enjoy the experience and app store ratings will decline. Once a request is sent, a response is expected to take anywhere from a few mins to an hour.
2. Manually verifying transcripts is time consuming, so by using Optical Character Recognition (OCR) to extract meaningful grade point data, an automated system can be developed to appraise a Tutor's transcripts.
3. This application should be simple and easy to use for all users regardless of competency with mobile applications. If an application is too complex for a user's needs, the in app experience is negative.
4. As with any other application, this application needs to be stable. Students and Tutors have restrictive schedules that do not allow for any kind of down time. Unstable applications lack professionalism which is unacceptable, especially when a payment system is involved.
5. The performance for this application is an important functionality that is measured by how responsive it is, how well it uses device memory and how well it uses device power. The users of the application should not experience any issues with the performance of the application.
6. This functionality would allow Students to provide a request for specific content to be covered during the Tutorial Session.

Bonus Information

Proposal Video: https://www.dropbox.com/s/zrbu16a8f6q7jo1/Proposal.mov?dl=0

Wireframe Video Walkthrough: https://www.dropbox.com/s/aceozw5hpql7ca0/wireframe.mov?dl=0

Wireframe PDF: https://www.dropbox.com/s/71ai544tujzzv7y/Wireframe.pdf?dl=0

dDevelopers
Satinder Singh – 100742575 / Atamjeet Singh – 100842821
Corey Cougle – 100745291 / Aarzan Irani - 100826631