Running head: Lab 1 – Tutor Dash Description

Lab 1 – Tutor Dash Description

Duncan J. Holterhaus

CS 411

Professor T. Kennedy

7 October 2019

Version 2

Table of Contents

List of Figures

List of Tables

1 Introduction

Tutor Dash will be a smartphone application aimed to improve the overall tutoring environment on campus. To achieve this goal, Tutor Dash is attempting to reduce the time of finding a tutor, give the tutee the ability to choose when and where they need a tutor, and give them the ease of mind that a tutor will be given to them without hassle. The application also is attempting to help tutors find their respective counterpart with a centralized location for them to locate a tutee on demand and get paid for their service.

Old Dominion University does not have the ability to supply tutoring to all students. The University does not offer tutoring sessions for all courses on the catalog and usually narrow it down to specific courses in a program. Additionally, most tutoring services end at seven in the afternoon, which creates an issue where tutoring is not offered when students are studying or doing assignments. Many students study at night, meaning that they can not use the walk-in hours of the university tutoring services when they need it the most.

Universities have a need for tutoring on their campuses. If students understand the material better, then the university's DFWI rates will improve significantly and the university can guarantee information retention. Currently, for students to find such help, they must navigate through multiple locations to find the exact course they're looking for. There is a need for a centralized location in the tutoring environment, so students do not waste time or get discouraged due to scattered, and sometimes conflicting, information.

There is an untapped market for students who have completed a course and want to tutor others. To offer their services now, they would either have to self-advertise or frequent tutoring websites that do not tailor themselves to the environment that that tutor is trying to appeal to.

This creates an issue where the scheduling and the payment of a tutoring session requires more effort than it should.

The current Old Dominion tutoring services are not enough. The information involving tutoring is spread across multiple sites and services, putting more barriers between the students and the help they require. Figure 1 shows the process of a current student seeking tutoring. The student can seek out help during the offered tutoring hours, which does not offer tutoring at night when most students are studying. This will force the student to find another tutoring source, often by looking online, at social media, or flyers across campus. The student must then find an available time to meet the tutor after verifying the tutor can teach the material the student is requesting. After the tutoring session, they must find a payment method that works for both parties.
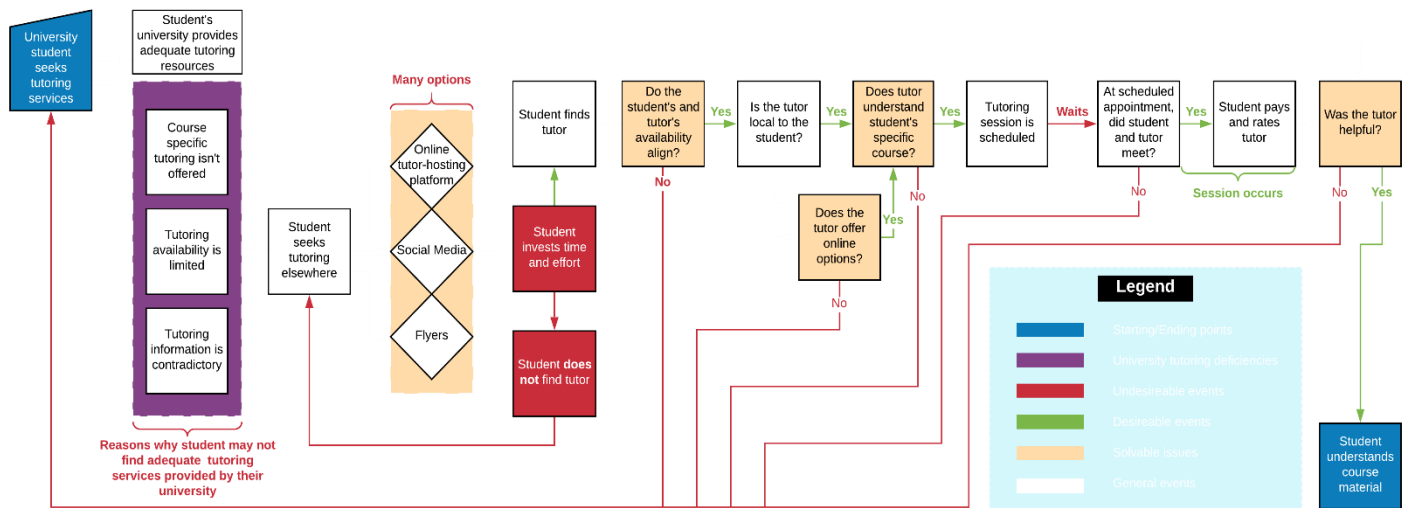
FIGURE 1 - STUDENTS SEEKING ACADEMIC ASSISTANCE

Students seeking to offer their services have a more difficult time in the current environment. Shown in Figure 2, the university student must create an advertisement showing their qualifications and post said advertisement on multiple platforms. The student must then wait for a response from a possible tutee. After a tutee response to the advertisement, they must find a location and a time for the meetup. After the session is complete and the tutee understands the material, they must then agree on a payment method.
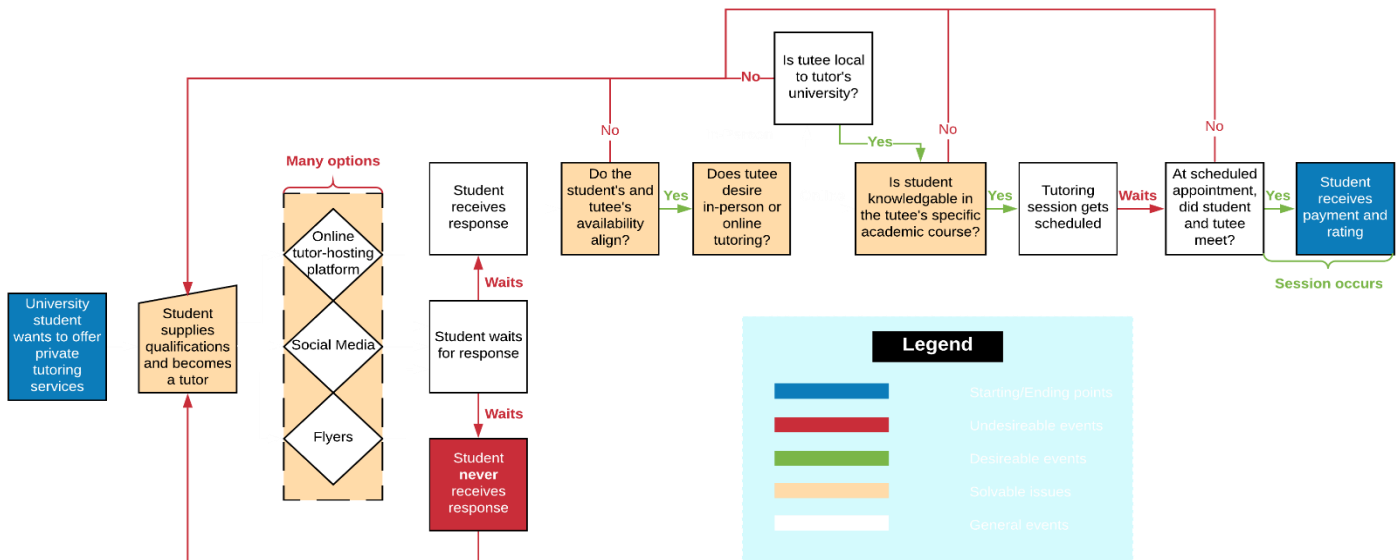
FIGURE 2 - STUDENTS SEEKING TUTORING ADVERTISEMENT

Tutor Dash will attempt to mitigate all problems explained. The smartphone application will attempt to create a centralized location for both market's needs. Tutor Dash will create an intuitive interface so the user can find their way around without trouble and increase the number of tutoring sessions that happen for any given university. The aim is to create a community that will use the up tapped market of student tutors and match them with students requiring help in their courses.

2 Tutor Dash Product Description

Tutor Dash's proposed solution is to create an application-based service that will unify

university students who are interested in tutoring and university students who desire course-

specific tutoring in real-time. This method will provide an extension to any university's current

tutoring services so that students can receive help for all courses. Tutor Dash does not aim to

replace the university's tutoring services, only to add more options to the tutoring market.

Tutor Dash will provide an intuitive UI that will point the users where they need to be

based on the type of user. There will be a section for each major junction in the process flow.

Tutors will have a screen specifically to find tutees in need, while tutees will have a screen to

find tutors offering help. If either party selects a counterpart, a chat window will be displayed

allowing them to communicate. This will allow them to set a location and a time for the session.

After the session, Tutor Dash will supply both parties with a payment method, so there is no

hassle in finding one.

For tutees, the process will flow from start to finish in a timely manner. For tutors, the

process flow will make finding tutees simple. Shown in Figure 1.3, Tutor Dash takes both

current processes and merges them into one. Instead of either user using multiple channels and

dialogue options, Tutor Dash will have every step explained previously as a separate screen for

the users to navigate. As shown in Figure 1.4, the account setup and the application usage will

have a simple process flow.

1. The user will download the application on their phone and fill in the basic user

   information.

2. If the user is a tutor, the account setup portion will request their transcript to verify that

   they can teach.

3. the user will be able to send out a notification to their respective counterpart.

4. The tutor and tutee will schedule a session in real-time, then the tutoring session will

    occur.

5. Once the tutee understands the material, the tutor will receive payment from the
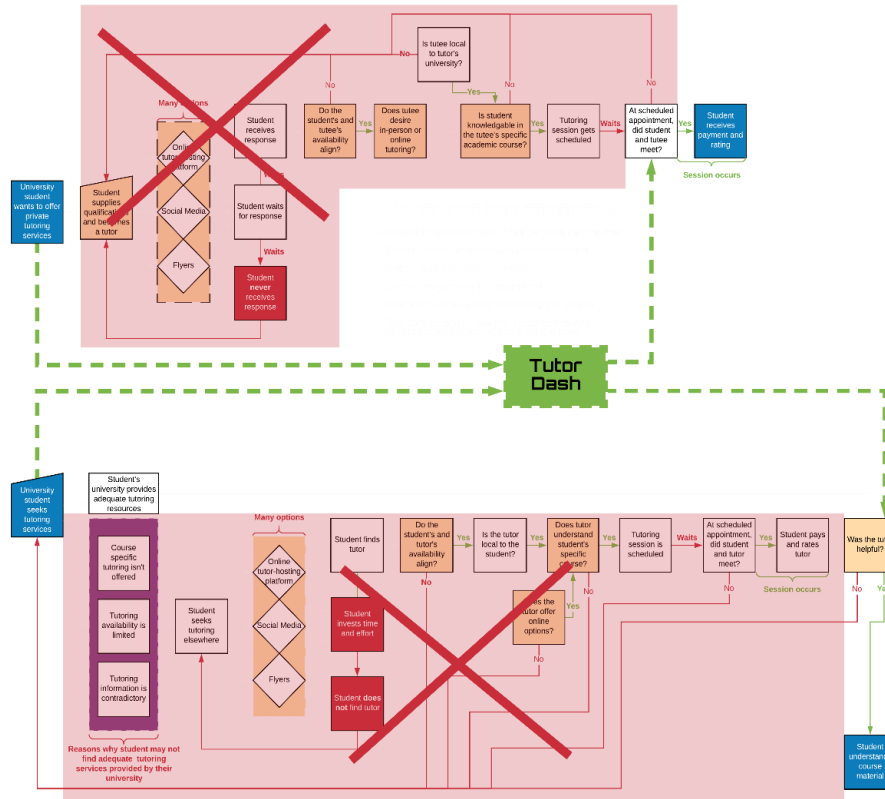
    application and both users will rate the other.

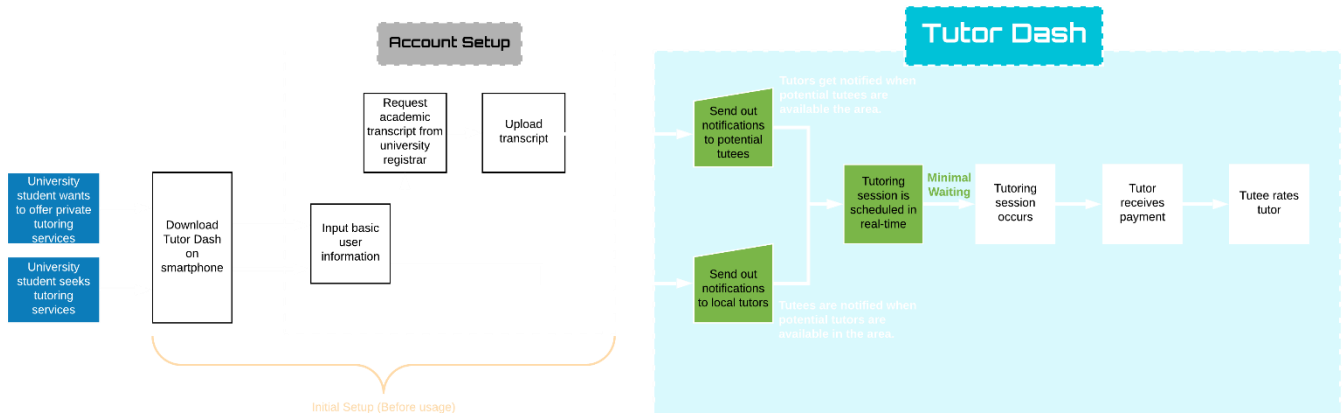**FIGURE 3 - HOW TUTOR DASH AFFECTS THE CURRENT PROCESSES**

FIGURE 4 - TUTOR DASH PROCESS FLOW

2.1 Objectives and Goals of Tutor Dash

1. Tutor Dash will attempt to the close the gaps that exist in the current private tutoring market. Most mediums currently used by possible customers do not offer the full picture, so users must use multiple forums to get their request out. This is not enough, and Tutor Dash aims to fill in those gaps.

2. Tutor Dash will create a centralized platform for all tutoring services. The scope of this goal goes beyond just Old Dominion University. Tutor Dash will attempt to fill the gap of current tutoring services on multiple campuses nationwide. Several Universities are expected for the real-world product; however, only Old Dominion University will be used for the prototype.

3. A thriving community is a very important goal for Tutor Dash. Without a customer base, all other functions will have no purpose.

4. The final goal is to mitigate or completely remove the stigma of having to receive tutoring. Reducing the friction of finding a tutor will greatly impact this goal.

2.2 Features and Capabilities

The user-base will be constrained to university students seeking to tutor and university students seeking tutoring. Tutor Dash will not attempt to expand into markets that do not involve some sort of official class being taught on a campus.

Every tutor that offers their services on Tutor Dash will need to go through a verification process to prove that they can teach the material that they wish to tutor in. This verification process will involve an official transcript that the user must request from the university. After the student has the official transcript, they will submit the file into the application and the application will gather the data automatically without any more user input.

A rating system will be implemented in order to keep the trust in both the tutors and the tutee reasonable. Both parties will be able to rate their respective counterpart and future users that receive or give services will be able to see the average rating that that individual has accumulated while on the application. This will be a secondary defense in case any of the tutors on the service find a way to cheat around the initial transcript submission. It will also provide the honest tutors with a means of gauging their interactions with future tutees.

2.3 Major Functional Components

Tutor Dash has three types of components: an entity, hardware, and software. An entity is any non-technological object that interacts with the system. Hardware is any physical technology that uses a software component. Software is any component that does not have a physical body and runs on the hardware components.

[This space intentionally left blank]

As shown below in Figure 2, software makes most of the Tutor Dash system. Multiple

systems run behind the interface that the user is presented, including algorithms and useful APIs.

Hardware and entities interact directly with the software and other entities in order to act as input

for the major functions.
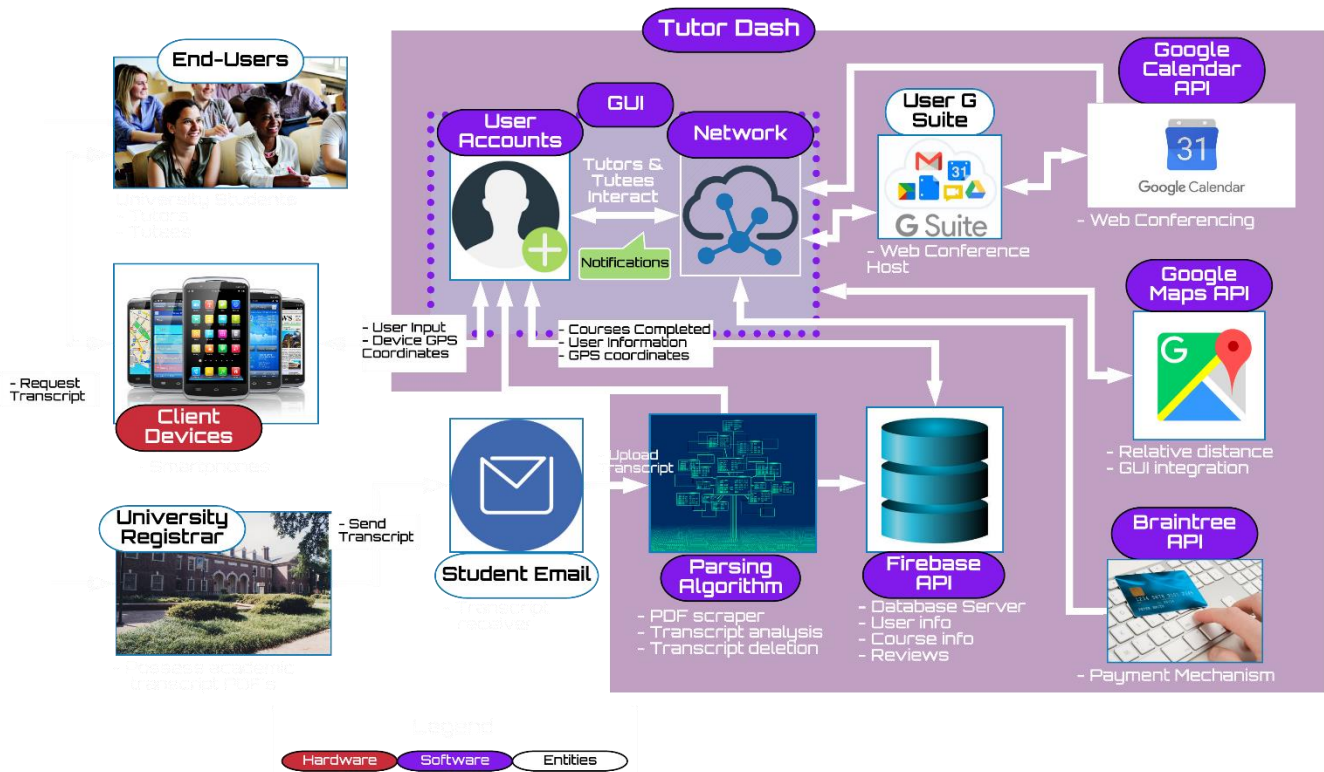
FIGURE 5 - MAJOR FUNCTIONAL COMPONENT DIAGRAM

2.3.1 Software

Software being utilized will include a simple to use GUI, user accounts, and networks

inside the application. Tutor Dash will also have a database run from Firebase and algorithms to

gather and produce information from user input. Multiple APIs will be utilized for specific

functions such as Google Calendar to set schedules, Google Maps to track and display student

locations, and Braintree to process in-application payments.

2.3.2 Hardware

Tutor Dash will utilize Android smartphones to run the application. The smartphone will allow the user to interact with the application and will display all user interfaces. Tutor Dash is expected to be viable on both Android and Apple devices for the real-world product.

2.3.3 Entities

The first entity will be the users who interact with the application directly. These users are expected to be university students and will be broken down into two groups: tutors and tutees.

Tutor Dash will indirectly interact with the university registrar. Tutors using the application will need to get their official transcript from their university registrar in order to prove that they can tutor a specific course. The user is expected to pay all fees involving this transaction.

The user will need to use their student email to receive their official copy of their transcript. The user will then take the PDF and submit it through a input function inside the application.

Tutor Dash will utilize the G Suite in order to schedule tutor sessions online. This entity will utilize the user's Google Calendar to provide a visual meetup time for both the tutor and the tutee. The G Suite will also be utilized to create a Google Meets session for users who wish to have a tutoring session over the internet.

3 Identification of Case Study

The Tutor Dash team has identified two major customer bases. First, the team will be looking for Old Dominion University students who are seeking tutoring from any source. This

will include students who are interested in using university resources and students who are interested in private tutoring services. Second, the team will be looking for Old Dominion University students who are seeking to tutor. This includes all students who have gotten an average or above grade on the classes they wish to tutor.

University students who are looking for tutees can come from two different backgrounds. One background would be an already established tutor that is looking to expand their clientele. This will include private tutors or tutors who have been advertising through other means. The second path would be an unexperienced university student who is looking to make some extra money through the application. These will include students who are trying to teach others in between classes.

A centralized location for tutoring is a must if the aim is to improve the overall tutoring industry for areas around universities. If tutoring services are improved, then benefits fall into place for both students and universities alike. Students will understand the material better which will improve their overall grades, reducing the number of failed courses and course retakes. The average amount of money students would be paying would drop because they will no longer have to pay for failed courses twice. Universities will also see an improve because more students will be passing with higher grades, leading to an improvement in the overall environment and rating of the school.

The last case study will be the projected future costumers of other universities. If the Tutor Dash team sees improvements as expected, meaning higher average grades, then most if not all universities will want to also see this improvement. This will open the door for Tutor Dash to assist these universities with their tutoring needs.

4. Tutor Dash Prototype Description

      The Tutor Dash prototype will be a small portion of the real-world product. The focus of creating this prototype will be to iron out the core functionality of the product so that the team can improve on the design before the non-essential portions are added. It will include a UI/UX for the user to view all the core functionality and to properly test the environment. It will also utilize the Firebase database to store all information regarding the prototype.

      The prototype will hold most of the real-world product functionality minus a few user management methods. The items that Tutor Dash will not have in the prototype will be free sessions as an advertisement mechanism, a night mode for night-time use, and support for multiple universities. Tutor Dash's prototype will also only have a partially functional automated pay rate calculator, partially functional reporting features, a partially functional refund system, and a partially functional blacklist method.

      The full list of functional items that will be shown in the prototype and not the real-world product can be seen in Table 1. Both the real-world product and the prototype will contain all data needed to store user information and functional information. This information will include relative distance between users, in-application transactions, weighted ratings for every course and user, and a verified university email for every user. The prototype will not have Night-mode, as this is a purely cosmetic function. The automated pay rate calculation will not be fully functional due to not having a lively costumer base, so stand in values will be used. Users in the prototype will be able to report other users, but no action will occur. Costumers will be allowed to request refunds due to poor performance, but this function will not be automated. Blacklist functionally will be implemented but will not be used in the prototype.

Feature	RWP	Prototype
On-the-fly tutor qualification based on transcript	Fully-Functional	Fully-Functional
University student verification based on email	Fully-Functional	Fully-Functional
Search results tailored based on tutor/tutee mode	Fully-Functional	Fully-Functional
Real-time scheduling	Fully-Functional	Fully-Functional
Weighted ratings for every course	Fully-Functional	Fully-Functional
Reviews and comments on user profiles	Fully-Functional	Fully-Functional
In-app payments/deposits (any transactions)	Fully-Functional	Fully-Functional
In-app messaging/history of conversations	Fully-Functional	Fully-Functional
Web conference and in-person meeting support	Fully-Functional	Fully-Functional
Relative distance user A is from user B appears in query	Fully-Functional	Fully-Functional
Night mode	Fully-Functional	Eliminated
Automated pay rate calculation for every course	Fully-Functional	Partially Functional - Mean & std. dev. of pay-rates will need to be mocked up
Reporting features	Fully-Functional	Partially Functional - Users can report, but no action will occur
Re-authentication when navigating back into app	Fully-Functional	Fully-Functional - However, this feature may disrupt the user experience
Refunds due to poor experiences	Fully-Functional	Partially Functional - Most likely, this will not be automated, but it still will exist
Free sessions/monetary bonuses	Fully-Functional	Eliminated
Blacklisting of users	Fully-Functional	Partially Functional - Capabilities will be implemented but not used
Support of multiple universities	Fully-Functional	Eliminated
Cross-platform support	Eliminated	Eliminated
Firebase console linked to test suite(s) with mockups	Eliminated	Fully-Functional

TABLE 1 - RWP VS. PROTOTYPE

The prototype aims to be a proof-of-concept for the real-world product. The core features for the prototype will include the verification and qualification algorithms for applying tutors, the user rating system, and the utilization of existing APIs to handle tutoring sessions via web conferencing. Algorithms that will be implemented will be the pay-rate calculation and the automated payment methods, to reduce the friction of payment.

4.1 Prototype Architecture

The architecture for the prototype will be very similar to the real-world product. The only change between the two will be the use of the Old Dominion University registrar for the official transcript. Both will use the same software shown in the MFCD, including the same algorithms and APIs. The prototype will utilize the student's official Old Dominion University email, but Tutor Dash will be expected to use the official email of any university for the real-world product.

4.1.1 Prototype Hardware

The only hardware the prototype will utilize is the end user's Android phone. Tutor Dash will not be available on Apple devices for the prototype.

4.1.2 Prototype Software

Tutor Dash will use established third party systems including Tutor Dash defined algorithms. The third-party systems being utilized will be a Firebase database server, the Android operating system, and Braintree. The algorithms being implements will include functionality pulled from third party APIs, a PDF transcript parser, a relative distance calculator, a pay-rate calculator, and a web-conference creator.

Firebase is a database server that will hold all important information. This will include all user information, such as login information, previous sessions, and payment methods. Firebase

will also include course information to help users find the courses they are looking to be tutored in.

The Tutor Dash application will be run on the Android operating system. This will handle all communication between the user and the application's functions.

Braintree will do all the application's payment processes. The payments will be done for both parties involved in a tutoring session. Tutor Dash will supply Braintree with the amount needing to be charged using an in-house algorithm.

Application algorithms will utilize third-party APIs to push and pull important data. These APIs include the Google Calendar API, the Google Maps API, and various Firebase APIs. Google Calendar will handle all session appointments for both users, including web-conference appointments. Google Maps will be used to display distance between possible tutors and tutees. Firebase APIs will be used to store and organize all data needed for Tutor Dash to function.

The PDF Transcript parser will be responsible for grabbing all available courses from the official university transcript and pushing passing courses to the user so that they may choose which courses to tutor. The parser will look for all available courses and store them in a data structure. It will then prune out all courses with a grade lower than the acceptable threshold and return the remaining values.

Tutors and tutees will need to know how far to travel to meet up with their respective counterpart. The application will use the Google Maps API to give both parties a rough distance between them so that they can choose the best location for a session. The relative distance calculator will aim to not get the location one hundred percent accurate so that users can avoid any nefarious shenanigans.

Tutor Dash will need to keep pricing competitive between users inside the application and outside the application. The pay-rate calculator will use multiple variables to keep the price high if the course is in demand or the price low if it is not. These factors also include the tutor's ability, the time of day, and course popularity.

Tutor Dash will have the ability to have tutoring sessions via web conferencing. The web-conference creator will use the Google Calendar API to create the session for both users so that they can have an easy to use communication method.

4.2 Critical Design Components

Tutor Dash will maintain three essential components. These components include the core database that will be run on Firebase, the UI/UX that will interact with the costumer, and the back-end algorithms that will calculate or generate the major interactions. The back-end algorithms will create the bridge between the database and the UI/UX so that the users can see applicable data on screen.

4.2.1 Database

Firebase will maintain all useful data for standard application functions. It holds a NoSQL, JSON, document-based database that is scalable to our user base and it stored in cloud storage. Firebase is being utilized due to previous experience by one of Tutor Dash's team.

Table 2 describes the data base schema for the Firebase database. The schema will include broad buckets such as users, school, courses, chat, blacklist, reviews, payments, and schedule. The user bucket will hold all information involved with the user. The school bucket will hold all the information regarding the school including courses offered. The courses bucket will hold the course name, the mean pay rate, and the standard deviation for that specific course.

The chat bucket will hold each individual chat log and the individuals involved in the chat. The blacklist will hold the email and ID of the individual the team marked as blacklisted. Reviews will contain the rating, comment, and timestamp of teach review, as well as the individuals that submitted the request and the individual being reviewed. Payments will hold both party's IDs, the amount and the datetime. Lastly, the schedule bucket will hold all information regarding a tutoring session.

[This space intentionally left blank]

User

UID
uName
fName
lName
email
picURL
schoolID
isTutor
isAvail
coursesOffered
coursesEligible
coursesPayRate
tutorRating
tuteeRating
inPerson
webConf
location
bio
timesSinceRequest
courseHours
courseID

School

schoolID
schoolName
schoolSuffix
courses

Courses

courseID
courseName
meanPayRate
stdDev

Chat

UID1_UID2
senderName
sendeeName
message
timestamp

Blacklist

email
schoolID

Reviews

UID
reviewerUID
rating
comment
timestamp

Payments

UID
receiverUID
dateTime
amount

Schedule

schoolID_UID
date
eventID
eventName
startTime
stopTime

TABLE 2 - DATABASE SCHEMA

4.2.2 UI/UX

The UI/UX will have four phases and two views, shown in Figure 6. The four phases will include the login/signup, tutor/tutee discovery, the session selection, and the active session. The two views will be the settings view and the user profile view.

[This space intentionally left blank]
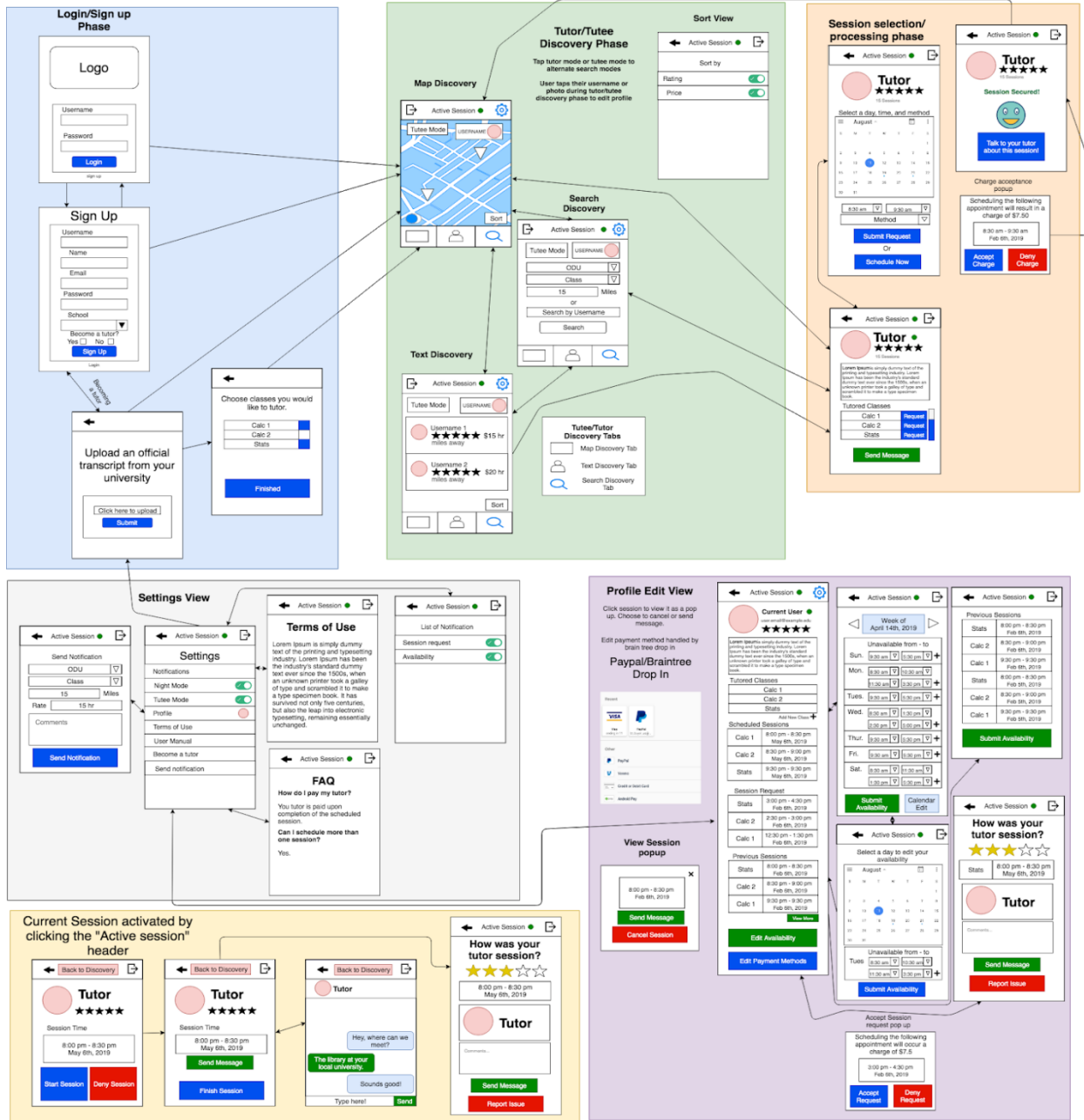
FIGURE 6 - UI/UX FULL IMPLEMENTATION

The login/signup phase, shown in Figure 7, will give the user the ability to either create their account if they do not already have one, or to sign into an existing account.
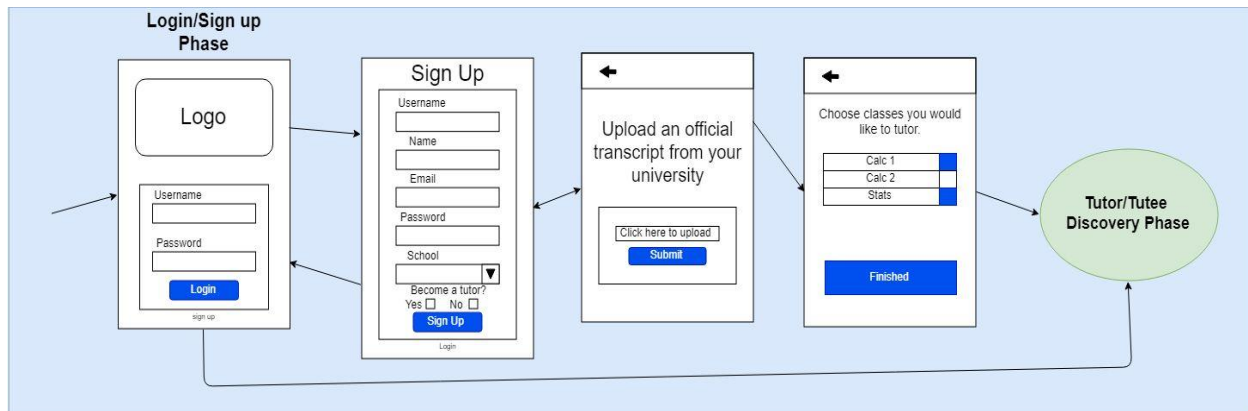
FIGURE 7 - UI/UX PHASE 1

The landing phase right after the login will be the Tutor/Tutee Discovery phase. Users will be able to switch their screen for whether they are looking for tutors or tutees. The list returned can be sorted on various aspects like courses taught and rating. There will also be an option to search people from a map on screen using the Google Maps API. If the users are looking for a specific tutor or tutee, they can search the database based on that individual's username.

[This space intentionally left blank]
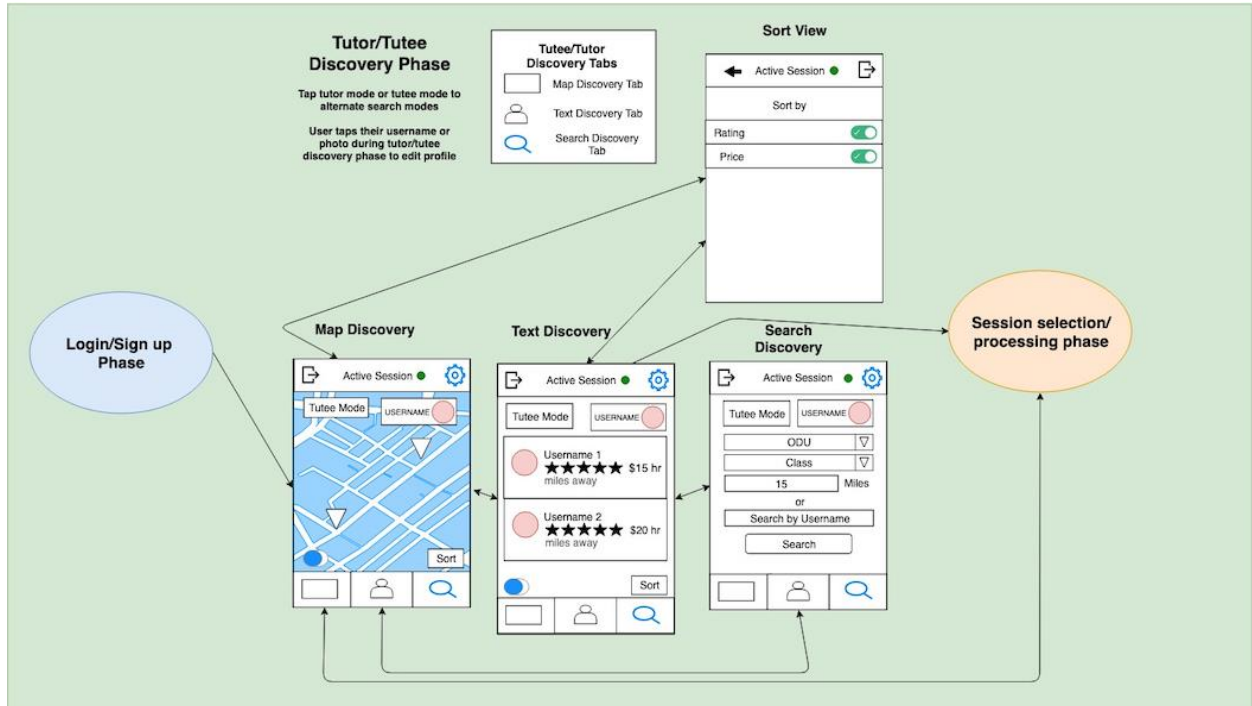
FIGURE 8 - UI/UX PHASE 2

With a specific tutor selected, the tutee can then see the Session Selection phase which will allow them to communicate with the tutor selected. As shown in Figure 9, this phase will also include the full page for the tutor's information. This page will include their username, rating, and custom biography. This view will allow the user to schedule a session during the tutor's available times or schedule a session in real-time.

[This space intentionally left blank]
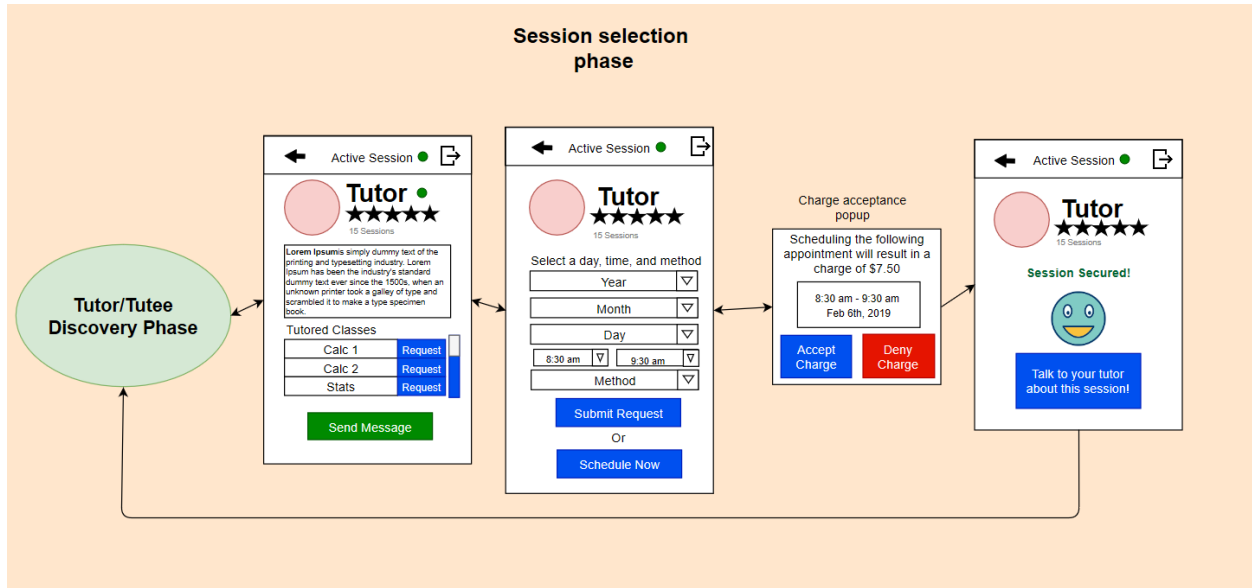
FIGURE 9 - UI/UX PHASE 3

When a session is scheduled, the application will show the user the Active Session phase, as shown in Figure 10. This will allow both users to communicate using an in-application chat box, as well as open and close an active session using this phase. When the session is closed, both users will rate each other and report their counterpart for anything against the user terms of service.

[This space intentionally left blank]
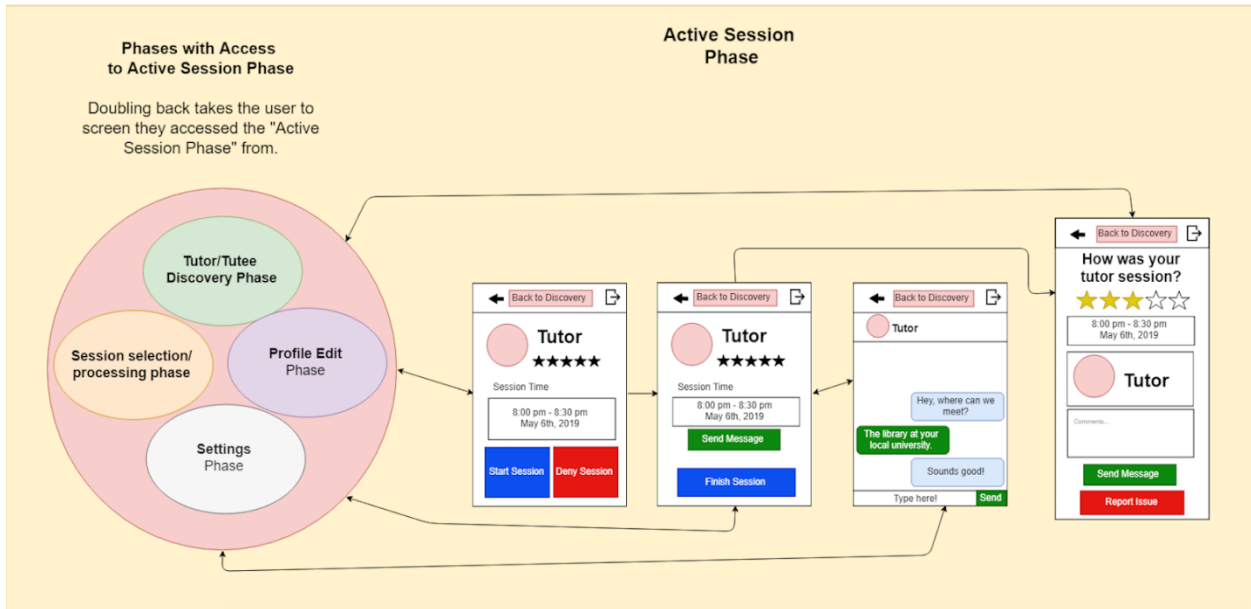
FIGURE 10 - UI/UX PHASE 4A

The active session will have an Active Session beacon, shown in Figure 11. This will allow the users to toggle in and out of their Active Session phase as this see fit.

[This space intentionally left blank]
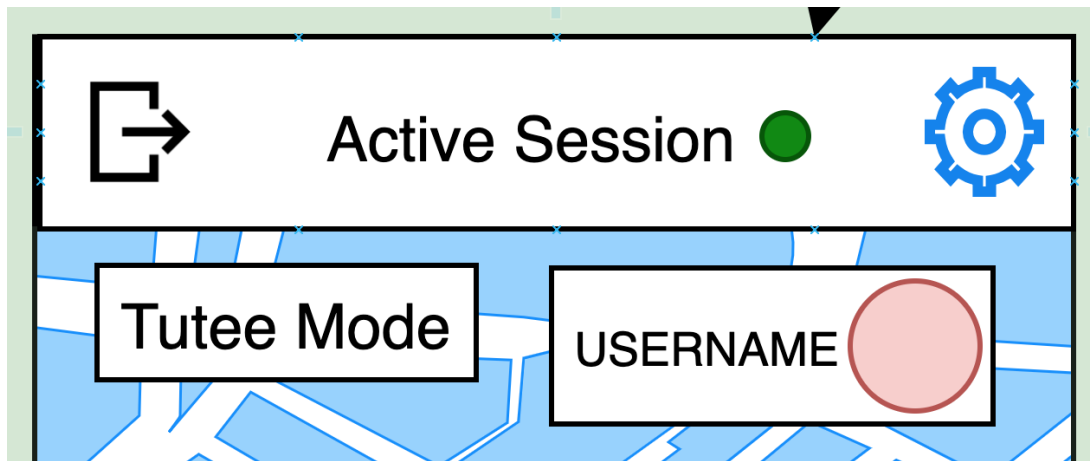
FIGURE 11 - UI/UX PHASE 4B

The users will also be allowed to adjust their personalize information using the two

views. As shown in Figure 12, the users will have access to their information through the settings

view. In this view, they can change their notification settings, view their profile, become a tutor

if they wish to teach, view their calendar, and review the terms of service and user manual.

[This space intentionally left blank]
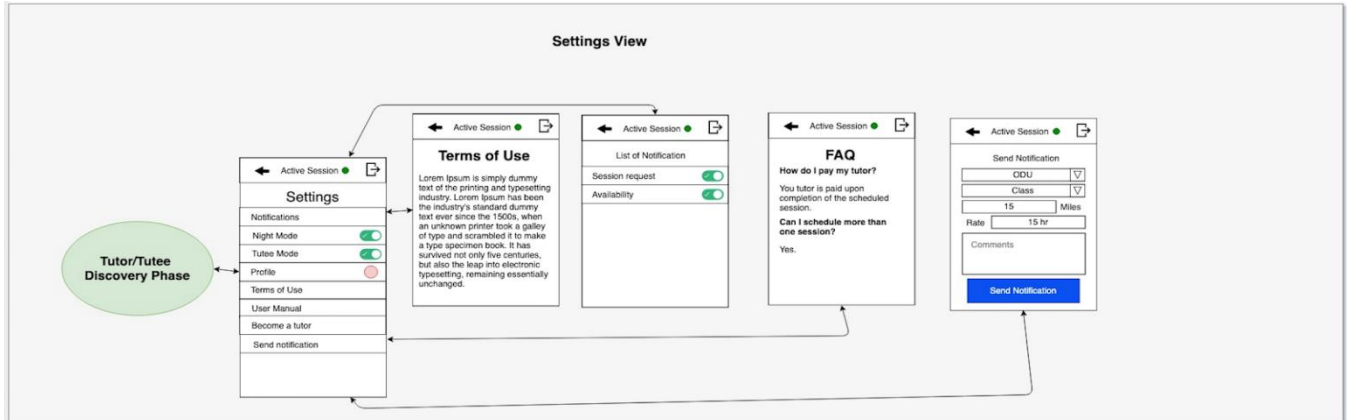
FIGURE 13 - UI/UX VIEW 1 - SETTINGS

Under the edit profile view, shown in Figure 13, the user will be allowed to adjust their personal information such as their username, e-mail, biography, tutored classes, future and past sessions, requests for sessions, and schedule. In this view, they will also be able to see their own ratings, based on reviews from users they have interacted with.
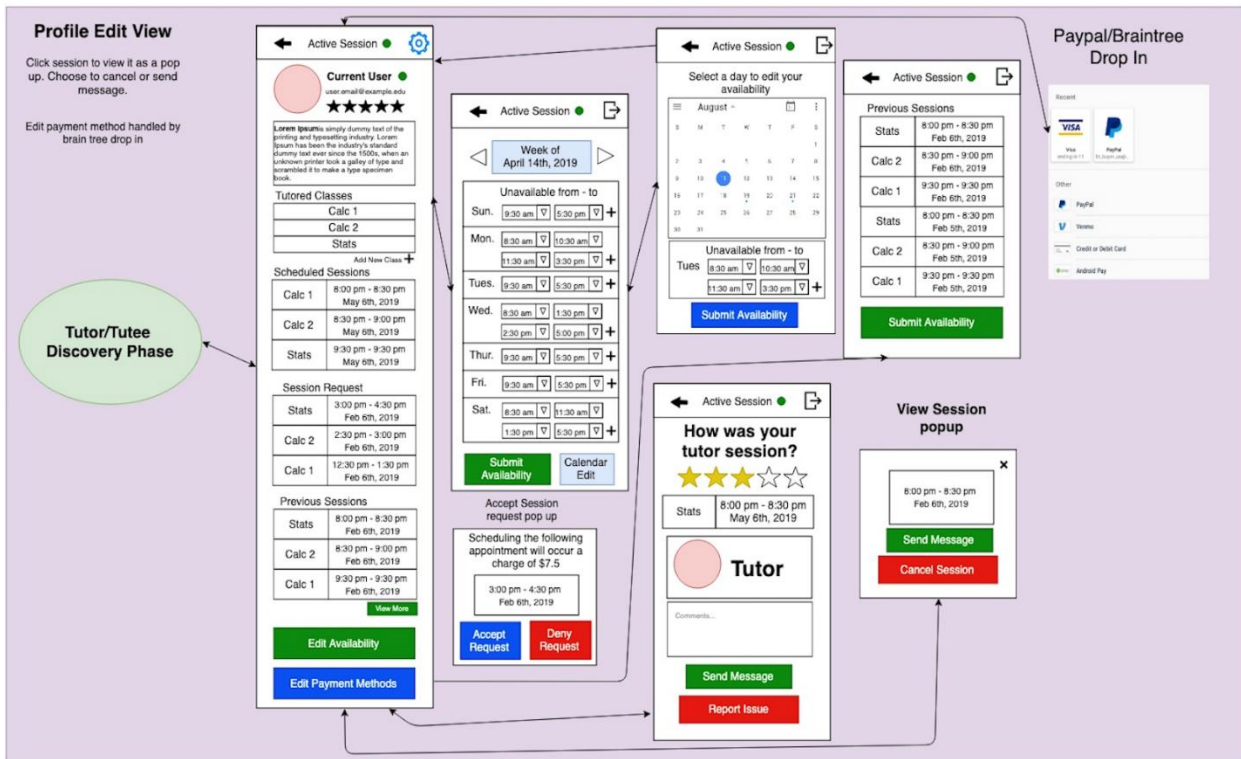
FIGURE 12 - UI/UX VIEW 2 - USER PROFILE

4.2.3 Algorithms

Tutor Dash will have five major algorithms: the PDF transcript parser, the relative distance calculator, the pay-rate calculator, the web-conference creator, and the basic payment logic.

The PDF transcript parser, shown below in Figure 14, will collect the user submitted official transcript and output eligible classes for the user to tutor. This algorithm will be utilizing the PDFBox Java library to read in the PDF file and parsing the string to find eligible courses based on a grade threshold.
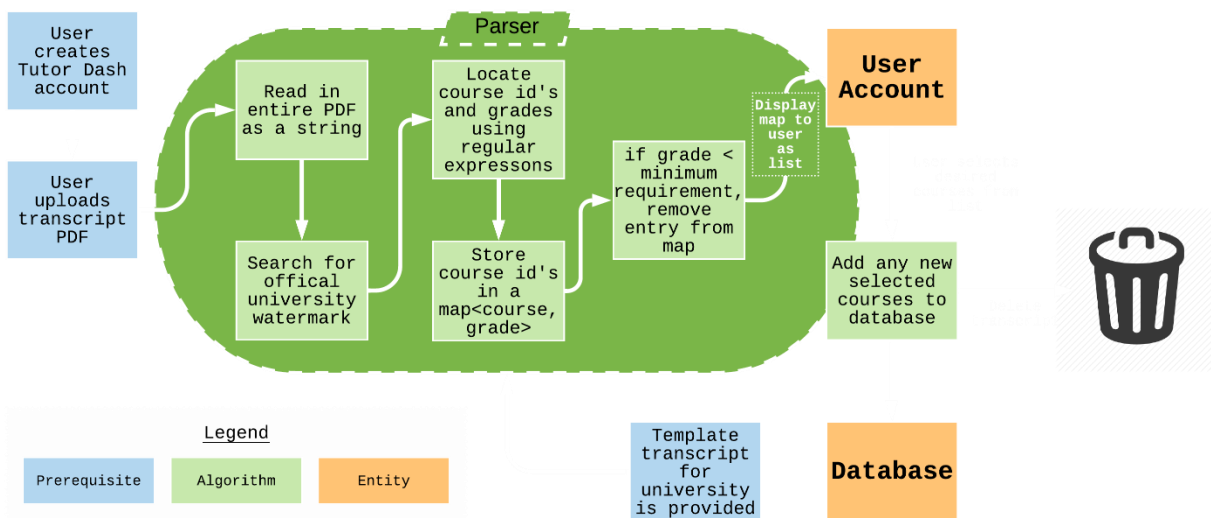
FIGURE 14 - PDF TRANSCRIPT PARSER ALGORITHM FLOW

The relative distance calculator, shown below in Figure 15, will collect both users GPS coordinates and estimate the relative distance between them. The application will poll the device's coordinates at a predetermined rate, and display this information using either the Google Map API or a text field.
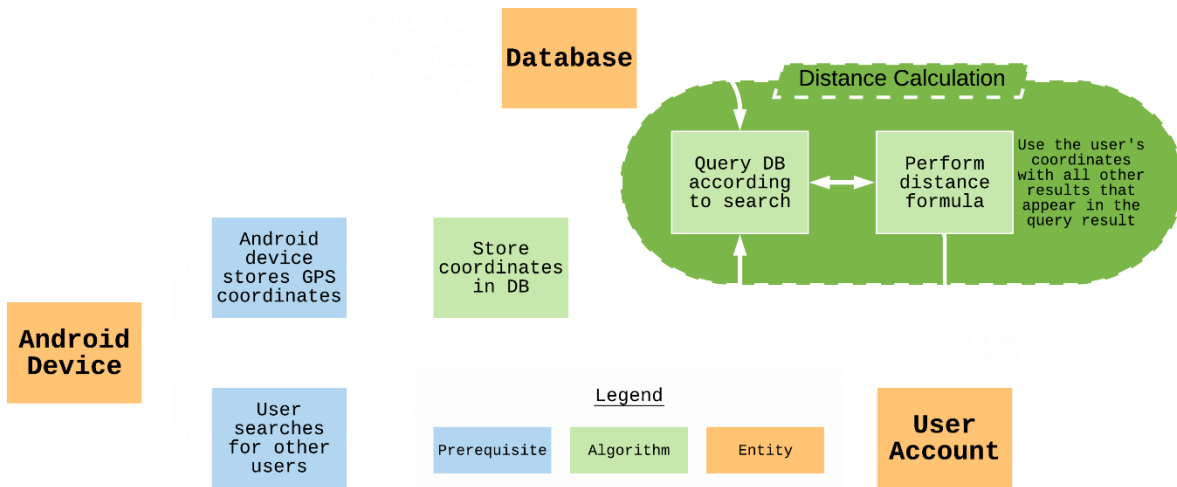
FIGURE 15 - RELATIVE DISTANCE CALCULATOR ALGORITHM FLOW

The application will use a pay-rate calculator, shown in Figure 16, that will use user information and course information to calculate the best fit price. The application will use an upper bound to keep prices from going above a competitive rate.
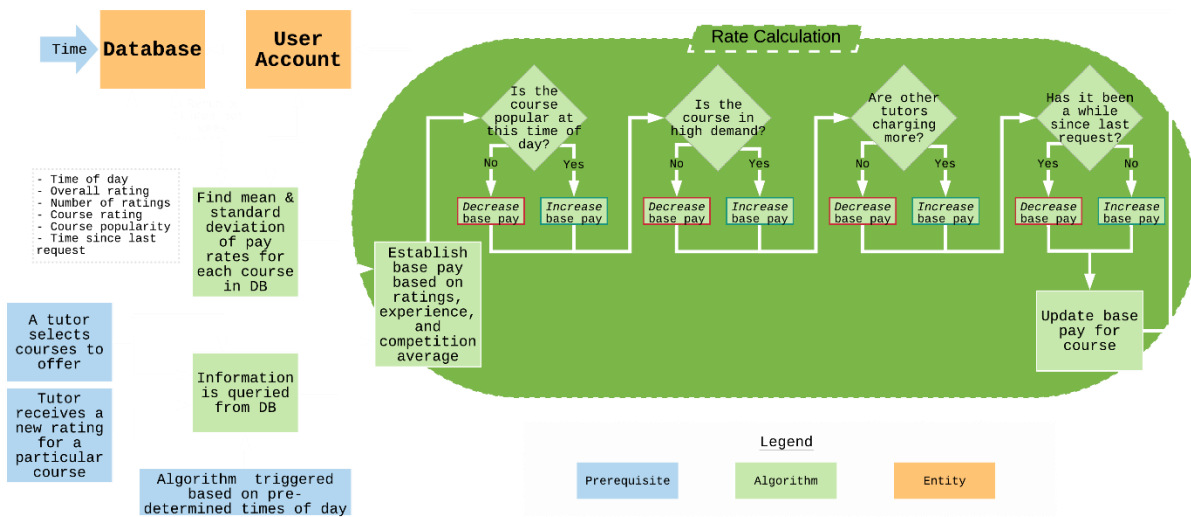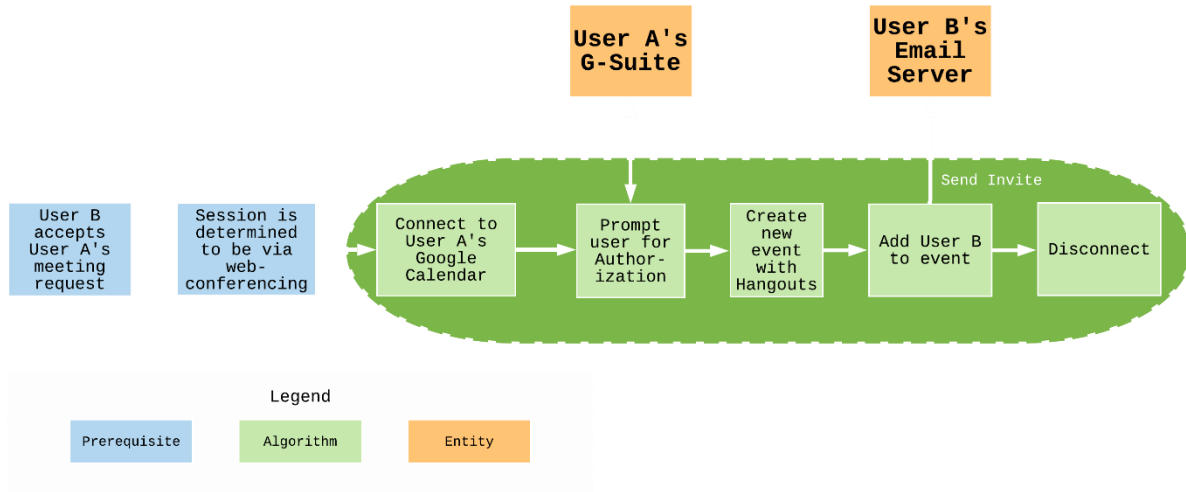

FIGURE 16 - COMPETITIVE PAY-RATE ALGORITHM FLOW

When a tutor and tutee want to use a web conference for the session, the application will

supply one using the web-conference creator, shown in Figure 17. This algorithm will pull from

the Google Calendar API in order to set up the session, and prompt both users for confirmation.

FIGURE 17 - WEB-CONFERENCE APPOINTMENT CREATOR ALGORITHM FLOW

The final algorithm will be a payment logic algorithm, shown in Figure 18, to make sure

that the application payments are working as intended.

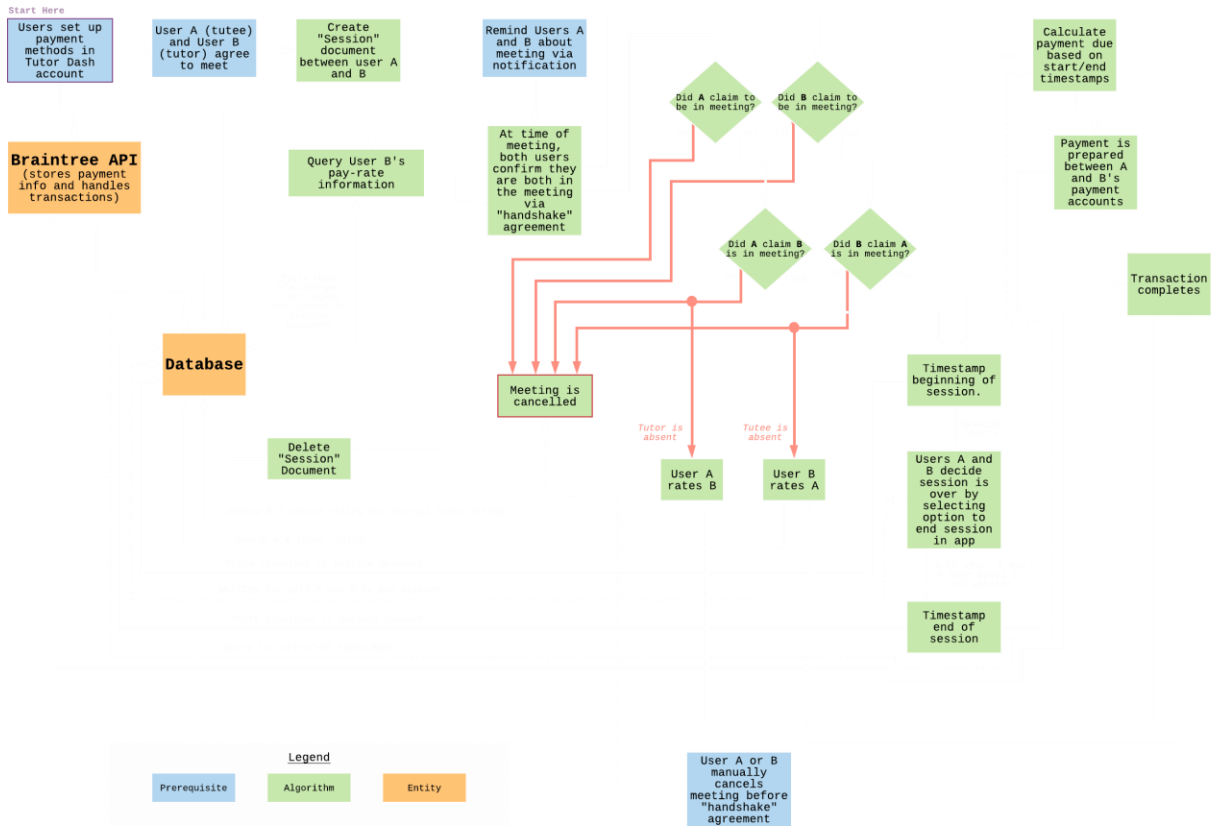[This space intentionally left blank]

FIGURE 18 - PAYMENT LOGIC ALGORITHM FLOW

4.3 Prototype Features and Capabilities

The Tutor Dash prototype will demonstrate an application that will direct tutors and tutees to their respective counterpart. A simple and easy to use payment method will allow both users to receive and send payments after a successful meeting, including the ability for users to meet via web-conferencing. If either user enjoys or dislikes their counterpart, a user rating system will be implemented so their opinion can be scored. Users will also be allowed to communicate through the in-application chat window, so there is no confusion on where the meeting will take place.

The aim of Tutor Dash is to create a centralized, understandable platform so that users can find what they are looking for in a tutor or tutee without hassle. If Tutor Dash meets at

prototype goals, then the application will have an intuitive design, easy-to-use tutor/tutee

discovery, and a method for spontaneous meetup.

4.4 Risks and Mitigations

Tutor Dash has narrowed down 21 possible risk factors that will be taken into

consideration during prototype development. Ideally, the application's rating system will weed

out any underperforming individuals to keep the trust factor in the application high. The PDF

transcript parser algorithm will be the first line of defense to verify that the user can tutor what

they say they can tutor. Google Calendar will prevent any users from double booking

appointments, and users can report other users if they see a violation in terms of service. All

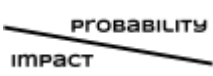suspected risks can be seen in Figures 19 and 20.

PROBABILITY / IMPACT	Very Low	Low	Moderate	High	Very High
Very High	T3, T4, L1	T6, L2		C3, C4	
High	T1, C7	C6, T9			
Moderate	C5	T8	C1, C8, T7		
Low	C2, T5		C10, T2		
Very Low		C9			

FIGURE 19 - RISK MATRIX OVERVIEW

ID	Risk Description	Mitigation(s)
C1	Student finds tutors to be unhelpful	• Rating system • Payment refunds
C2	Prospective tutors faking their qualificaitons	• Require official transcript from university registrar • Make tutors only eligible to tutor classes he/she has received a B or higher in
C3	Shortage of tutors	• Give small bonuses to tutors for a limited time (similar to Uber's business model)
C4	Shortage of tutees	• Give free sessions to new users • Give loyaltee-free sessions for a certain number of usages
C5	Tutor/tutee leaves a false negative review	• Allow users to challenge reviews (requires manual investigation) • Withhold ratings/reviews until both users agree on justification
C6	Users abuse application; use app maliciously	• Require users to agree to the terms of use agreement • Blacklist (ban) users who violate the terms of use agreement
C7	Identity theft; non-users impersonate users and/or users impersonate other users	• Re-authentication when navigating to app from outside window • "Handshake" agreement between users when sessions begin
C8	Participating tutor/tutees don't show up to their scheduled meetings	• Preallocate payments • Require deposits • Threat of poor ratings
C9	Users try to book overlapping sessions	• Only allow users to make appointments for times they don't currently have a scheduled session • Applies to both tutors and tutees
C10	Tutors are not adequately prepared to engage with tutees via web conferencing	• Alert users of the minimum requirements for web conference meetings upon selecting 'web conferencing' as a tutoring preference.
T1	Payment is not received	• Integrate usage of a 3rd party API designed to handle e-transactions • Braintree
T2	Difficulty automating the process of reading a submitted transcript	• Define reusable code for general case • Optimize as more information is discovered
T3	Database server failure	• Use reliable servers maintained by large corporations • Firebase
T4	Security breach	• Use 3rd party APIs which are already secure
T5	Application is not compatible on all android devices	• Define minimum SDK for weaker hardware phones • Define normal SDK for normal hardware phones
T6	Network server failure	• Server redundancy
T7	Pay-rate algorithm doesn't calculate competitive rates	• Determine a base pay that will increase/decrease due to various factors • Compare pay-rates of similarly rated tutors who tutor the same courses

ID	Risk Description	Mitigation(s)
T8	Web-conferencing session is not set up properly	• Use Google Hangouts • Ask permission from user to schedule events in their calendar via Google API's
T9	Unexpected interruption prohibits online sessions from occurring	• Refund payments in this case as long as both parties arrived to the meeting. • Use a Google Hangouts, a commonly used web conference tool maintained by a large corporation.
L1	Violating The Family Education Rights and Privacy Act (FERPA)	• No portal access • Transcripts are analyzed then thrown out • Users agree to grade disclosure in terms of use agreement
L2	Users use application for illegal activities	• Terms of use agreement • Reporting features

FIGURE 20 - RISK MATRIX DESCRIPTIONS

4.5 Prototype Objectives and Goals

Tutor Dash will push towards five functional goals and objectives.

1. An easy to understand UI that will not confuse users.

2. Maintain the ability to find a tutor or tutee in a timely manner.

3. A respectable payment system that does not charge anything other than what the users expect.

4. Maintain multiple pathways for users to communicate and have sessions.

5. A working rating system so users can self-regulate their community.

4.6 Prototype Development Challenges

The Tutor Dash team will face many challenges during the prototyping process. Most of the Tutor Dash team is inexperienced in the Android application development. We also have no experience in any of the APIs that will be implemented. The team will also have to develop code to handle payments, which can create problems with users if said code fails. Another issue is maintaining a competitive rate that can stay competitive within and outside the application. Finally, doing all these processes under time constraint.

5 Glossary

Course-specific tutoring: Academic assistance services provided for a particular course at a particular university.

DFWI: An acronym for Drop/Fail/Withdraw/Incomplete.

DFWI rates: An abbreviation for Drop/Fail/Withdraw/Incomplete rates. Represents the ratio of university students who do not complete their courses to students who do complete their courses.

Direct Competitor: Another product or company which is solely involved in the same domain space as Tutor Dash.

Entity: A person, object, or external server that serves as a leveraged functional component of the Tutor Dash product.

FERPA: The Family Educational Rights and Privacy Act is a United States federal law that protects the privacy of educational records.

G Suite: An API created by Google to access all their services.

Indirect Competitor: Another product or company which is involved in the same domain space as Tutor Dash, but doesn't not focus solely in that space.

Tester: Individuals responsible for testing the quality of the software.

Tutee: A university student seeking academic assistance.

Tutor: A university student offering independent tutoring services that are qualified based off previously taken courses.

Uber: A ride-hailing company that offers the Uber mobile app, which you can use to submit a trip request that is automatically sent to an Uber driver near to you, alerting the driver to your location.

6 References

"Academic Tutoring in Comprehensive Universities." Hanover Research, 2014. URL:

> https://www.hanoverresearch.com/wp-content/uploads/2017/08/Academic-Tutoring-in-

> Comprehensive-Universities.pdf

"Academics." Old Dominion University, 8 Jan. 2019. URL: www.odu.edu/academics

"Campus Tutoring." Old Dominion University, 19 Jan. 2019. URL:

> www.odu.edu/success/academic/tutoring#tab125=0

"Course-Specific Tutoring." Old Dominion University, 19 Jan. 2019. URL:

> www.odu.edu/success/academic/tutoring/course-specific

"Courses of Instruction." Old Dominion University, Feb. 2019. URL: catalog.odu.edu/courses/

Ciscell, Galen, et al. "Barriers to Accessing Tutoring Services Among Students Who Received a

> MidSemester Warning." ERIC, Pacific Lutheran University - Department of Sociology,

> 2016. URL: files.eric.ed.gov/fulltext/EJ1114513.pdf

Evans MDR, Kelley P and Kelley J (2017). Identifying the Best Times for Cognitive Functioning

> Using New Methods: Matching University Times to Undergraduate Chronotypes. Front.

> Hum. Neurosci. 11:188. doi: 10.3389/fnhum.2017.00188. URL:

> https://www.frontiersin.org/articles/10.3389/fnhum.2017.00188/full?&utm_source=Emai

> l_to_authors_&utm_medium=Email&utm_content=T1_11.5e1_author&utm_campaign=

> Email_publication&field=&journalName=Frontiers_in_Human_Neuroscience&id=2394

> 92

"Facebook - Groups." Facebook Help Center, Facebook, 2019. URL:

> www.facebook.com/help/1629740080681586?helpref=hc_global_nav

"Find a Local In-Home Tutor Today." HeyTutor, HeyTutor LLC. URL: heytutor.com/

Fry, Natalie. "New Research Reveals That College Students Study Best Later in the Day."

NevadaToday, University of Nevada, Reno, 11 Apr. 2017. URL: www.unr.edu/nevada-

today/news/2017/best-time-of-day-to-study

"Peer Assisted Learning" BMC Education, 8 March 2006 URL:

https://bmcmededuc.biomedcentral.com/articles/10.1186/1472-6920-6-18

Pierce, Dennis. "Supporting Students Beyond Financial Aid", 2016 URL:

http://eds.b.ebscohost.com.proxy.lib.odu.edu/ehost/detail/detail?vid=0&sid=d93df6c4-

3729-4b62-8d58-

95e25c309878%40sessionmgr102&bdata=JnNpdGU9ZWhvc3QtbGl2ZSZzY29wZT1za

XRl#AN=114789419&db=ehh

Qayyum, Adnan. "Student Help-Seeking Attitudes and Behaviors in a Digital Era." International

Journal of Educational Technology in Higher Education, vol. 15, no. 1, 2018, doi:

10.1186/s41239-018-0100-7. URL:

https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-018-

0100-7

"Skooli Tutors Online." Skooli Online Tutoring, Skooli, Feb. 2019. URL:

www.skooli.com/prices/students

"Student as Peer Tutors" BMC Education, 9 June. 2014. URL:

https://bmcmededuc.biomedcentral.com/articles/10.1186/1472-6920-14-

115https://bmcmededuc.biomedcentral.com/articles/10.1186/1472-6920-14-115

Topping, J. Keith. "Trends in Peer Learning", 19 Jan 2007 URL:

https://www.tandfonline.com/doi/full/10.1080/01443410500345172?scroll=top&needAc

cess=tru

"Tutor Matching Service - How It Works." Tutor Matching Service, Tutor Matching Service,

     2019. URL: tutormatchingservice.com/#/about

"Tutor.com - The Princeton Review." Tutor.com, The Princeton Review, URL: www.tutor.com

"Tutors - Care.com." Care.com, Care.com, Feb. 2019. URL: www.care.com/tutors

"Wyzant." Wyzant Resources, Wyzant Inc., Feb. 2019. URL:

     www.wyzant.com/howitworks/students