Lab 1 – Tutor Dash Product Description

Edwin R. Ordona

Old Dominion University

CS 411W

Professor T. Kennedy

October 7, 2019

Version 2

Table of Contents

List of Tables

List of Figures

Lab 1 – Tutor Dash Product Description

1. Introduction

For university students, tutoring sessions between peers can be beneficial to both parties as they help reinforce content taught in their courses, which in turn helps alleviate DFWI rates by bolstering major retention [11]. The availability of these sessions, however, can sometimes be unpredictable without browsing through several resources provided by campuses or third-party services.

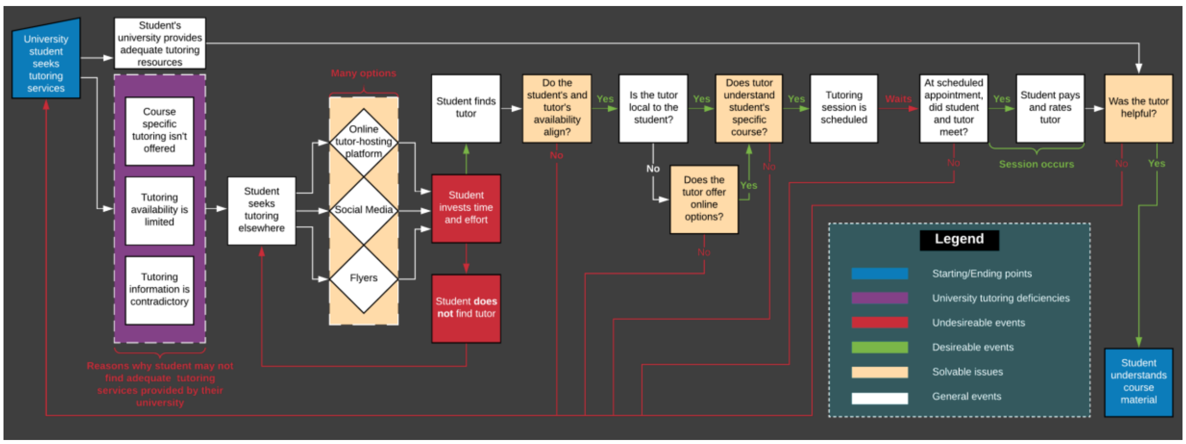

Figure 1 - Students Seeking Academic Assistance

In Figure 1, tutees are reliant on the availability, location and skillsets of tutors. Qualified tutors that want to provide their own services officially for previously taken courses may have to rearrange their schedules to fill in a requested time-slot requested by their tutees, which may not be convenient. If the tutors in question are seeking monetary compensation, there is no centralized platform that handles transactions between tutors and tutees in an appropriate manner.

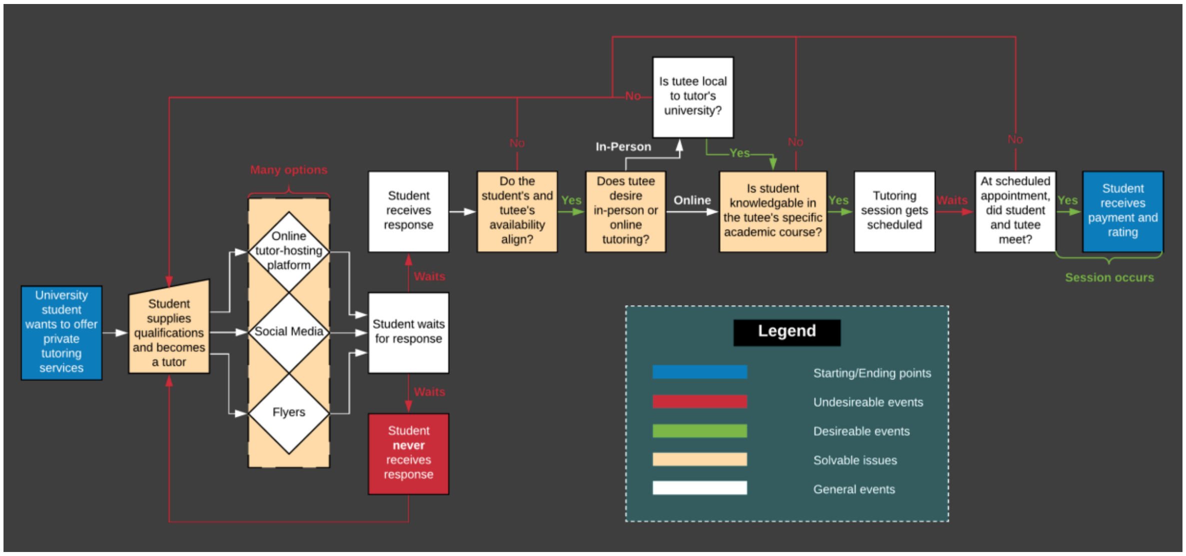

Figure 2 - Students Seeking Tutoring Advertisement

There are several problems tutors and tutees face when trying to provide or seek out educational services. Tutoring resources provided on campus may not be sufficient enough for a variety of reasons. For instance, students may not be able to allocate enough free time towards tutoring services due to work or other personal obligations. In some cases, the free time they are able to allocate may not match up with the availability of the tutors. These resources are often divided by department and are not easily accessible, causing students to overlook these services. Tutees looking for course-specific services may have difficulty finding assistance as tutoring sessions often cover broad subjects instead of being tailored to one specific course, resulting in tutees needing to direct their attention to the private market. The reliability of private tutors can vary as their services may not be specifically designed on a course-by-course basis. Tutors marketing their services so that students seeking help for specific courses can prove to be difficult as there is currently no centralized network for them to advertise themselves efficiently as depicted in Figure 2.

Tutor Dash will be a mobile tutoring hub that aims to connect university students with each other by managing the exchange of tutoring services. The application will be designed to act as a supplement for existing educational services provided by campuses and will benefit all students that seek out or provide educational assistance, while also having support for tutors seeking monetary compensation.

2. Tutor Dash Product Description

Tutor Dash will be an innovative, Android-based solution that will serve as a way to mitigate how much time is spent by tutors seeking tutees and vice versa by streamlining the current process.

Currently, both tutors and tutees have to go through multiple steps to ensure a successful tutoring session. Factors include the availability of both parties, the physical distance between each party, and whether or not the scheduled session actually occurred. For tutors, advertising their services may not always elicit a response if potential tutees are never exposed to them. After a tutor receives a response, there are certain checks which must be done. If at any point a party does not meet the requirements for a session be scheduled, time is wasted and the cycle to find another potential session continues.

[This space intentionally left blank]

Tutor Dash automates the processes for tutors and tutees through the consolidation of available resources.

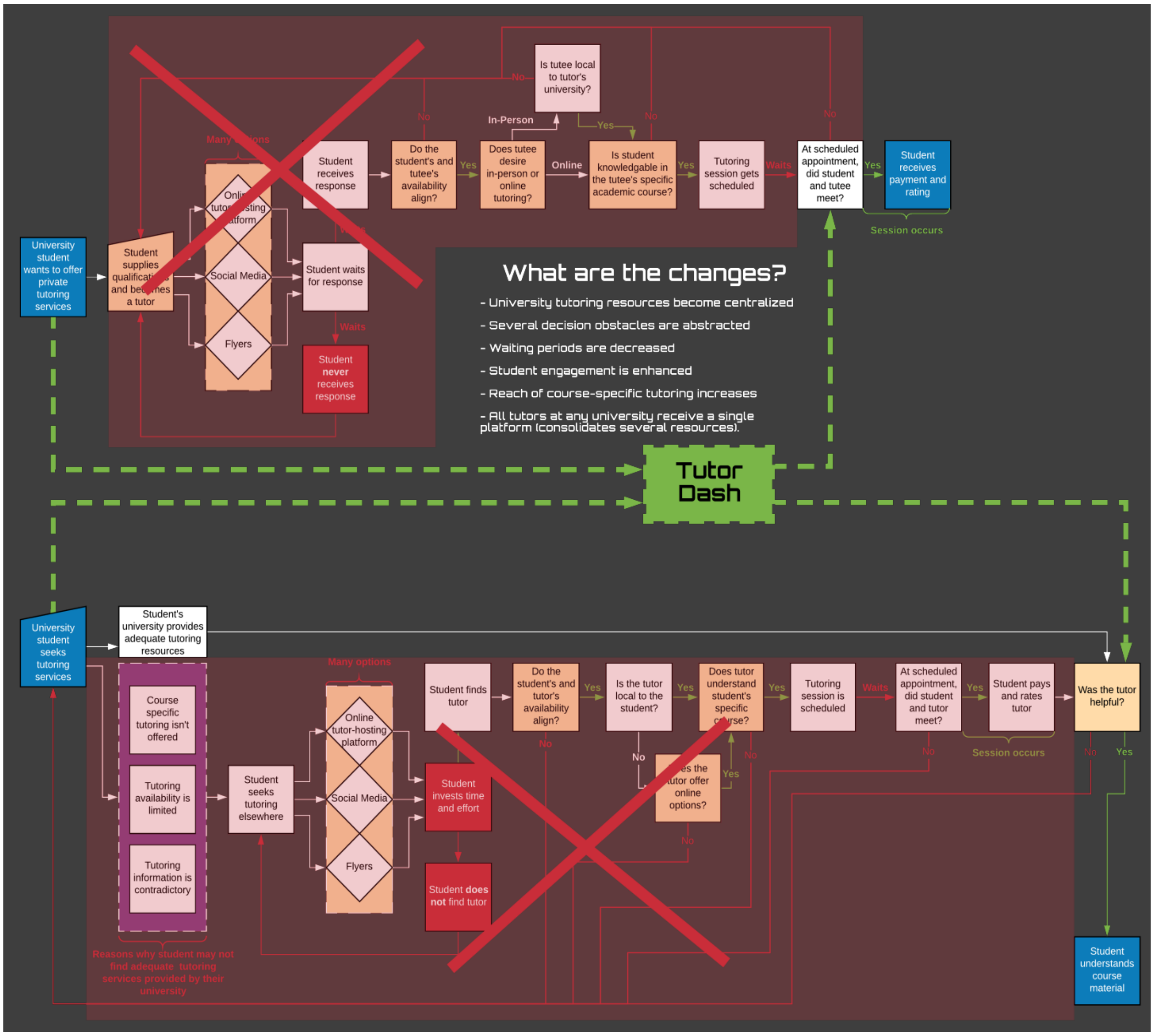

Figure 3 - How Tutor Dash Affects the Current Processes

The application centralizes the processes that both tutees and tutors need to go through to take part in a tutoring session and eliminates a majority of the decision obstacles that wastes unnecessary time for both parties involved. It will be a niche tool that promotes peer-assisted learning by connecting students with each other in a straightforward manner, similar to how Uber connects drivers with people looking to travel to a certain destination.

Tutors and tutees will need to download Tutor Dash on a mobile device. For tutors, an extra step requires them to upload a transcript to determine which courses they can provide services for. Tutor Dash then sends notifications to tutors that are seeking potential tutees and vice versa, automating the search process. The application then facilitates the meetings, handles any transactions between the tutor and tutee, and provides a rating system to help increase the quality of service for future and returning users. Figure 4 visually outlines the process flow that is simplified by Tutor Dash.

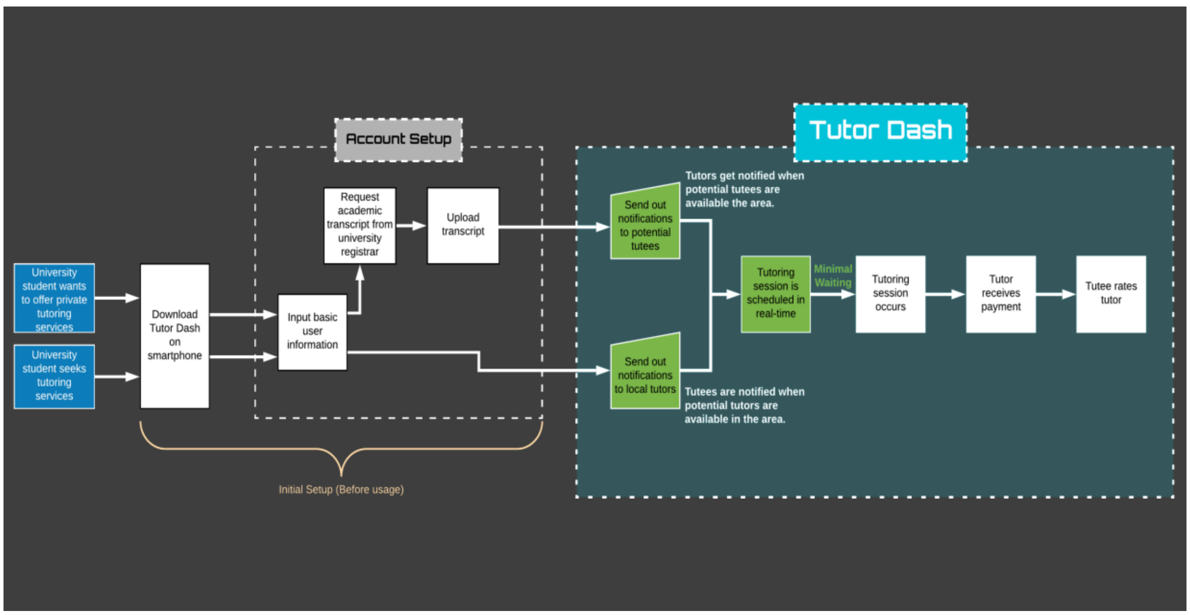

Figure 4 - Tutor Dash Process Flow

2.1 Objectives and Goals of Tutor Dash

For Tutor Dash, the overall goal is to provide university students a centralized platform that consolidates existing tutoring resources provided by campuses with those in the private market. By initially limiting the scope to Old Dominion University students, the development team will be able to build and develop a customer base that will eventually expand the application's reach to multiple universities. Achieving this would help students be exposed to tutoring services that they need from their peers while simultaneously helping those who wish to provide said services. Long-term effects include the acceptance of tutoring as a whole, as the negative stigma surrounding tutoring (such as tutoring sessions not being helpful enough or convenient time-wise) affects the number of students that actively seek out help from fellow students to assist in their courses.

2.2 Features and Capabilities

Tutor Dash is a consolidation of already existing features provided by other services, with the exception of the user-base being strictly comprised of university students. Tutors are qualified based on course performance and will be automatically handled by the automatic PDF transcript parser. A rating system will be implemented to allow tutees to determine if they will receive satisfactory treatment from potential tutors. Tutors will also benefit from this rating system by being able to advertise their services efficiently, while also allowing them to filter out clientele they do not wish to work with. An availability toggle will ensure that users will only get notified if they are looking to provide or receive tutoring services. Because of the nature of the application, there is flexibility in what can be tutored, which ranges from broad subjects to course-specific assistance. The service is also available at all times, which can be beneficial to students who cannot access official resources due to scheduling conflicts. Pay rates are automatically calculated by algorithms to ensure a competitive rate, and all transactions are securely handled by an external API. Lastly, Tutor Dash will schedule meetings of any type, whether it is online or in person.

2.3 Major Functional Components

The Tutor Dash application will consist of several major functional components that will ensure a quality experience.

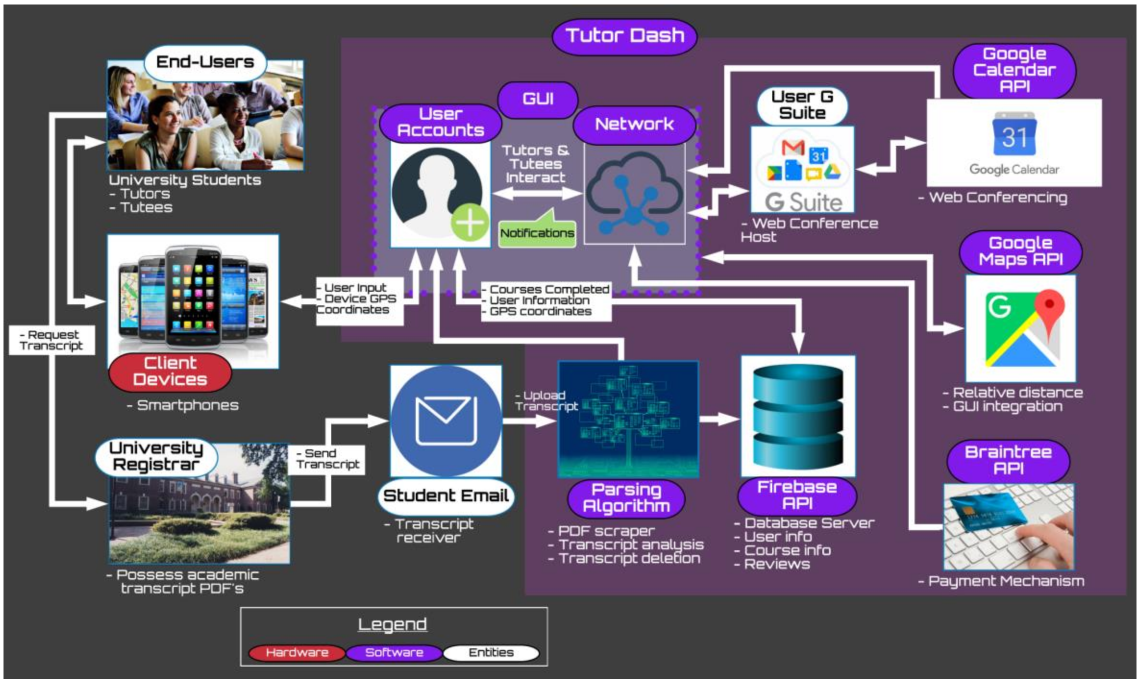

Figure 5 - Major Functional Component Diagram

2.3.1 Software

Tutor Dash is a software package that utilizes different APIs and technical coding to help define itself as a centralized, tutoring platform. Students will utilize the application's features with a general user interface (or GUI) that will simplify student-to-student communications and the various connections used by the Tutor Dash network. The GUI will also trigger relative notifications to ensure active engagement between tutors and tutees. User accounts will be unique to each student to ensure that returning users will have access to any data they require from the application. The database (which will be developed using Firebase) will house all user and course information, as well as the student ratings. The parsing algorithm scrapes and analyzes PDF files for pertinent information, and then it deletes the transcripts to prevent the mishandling of sensitive data. The Google Calendar API will handle the scheduling of online sessions. The Braintree API will handle all tutor-related transactions. Lastly, the Google Maps API will be used to calculate relative distance and will be integrated seamlessly into the application.

2.3.2 Hardware

End-users will only need a compatible Android device with an internet connection via WI-FI or mobile network to access the Tutor Dash application. No other devices are required.

2.3.3 Entities

The entities are comprised of university students (both tutors and tutees), the university registrar (to acquire official transcripts), student emails (to retrieve transcripts from the university registrar) and the users' G-Suites (to host online tutoring sessions).

3. Identification of Case Study

To summarize the overall problem described in Section 1, existing tutoring services are limited in how can be accessed. Tutor Dash will be developed with university students as its primary market. Tutors who want to provide tutoring services and tutees who want assistance for course-specific subjects will find Tutor Dash to be an exceptional supplement to official resources provided by campuses.

The centralization of resources within Tutor Dash will benefit the university as well since tutoring (via peer-assisted learning) has proved to increase course and major retention rates. Tutees who utilize the application can help save money by passing their courses (so they do not have to spend extra money to retake them) and finding acceptable tutoring rates. Tutors can gain tutoring experience as well as compensation for their services.

Future customers include students from all campuses. By developing and containing the initial product to a single university, the initial number of users will continue to grow and through word of mouth, the application's scope will spread outwards.

4. Tutor Dash Prototype Description

The Tutor Dash prototype will be developed as a proof-of-concept for the real-world product. It will implement Firebase to handle database operations, a basic user interface (UI) and user experience design (UX) for front-end operations, and several algorithms for back-end functionalities.

For the purpose of the prototype, several functionalities from the real-world product be omitted or partially implemented. All discernable differences are listed in Table 1.

Table 1 - RWP vs. 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

Table 1 – RWP vs. Prototype (Cont.)

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	Eliminated - Unnecessary for prototype
Support of multiple universities	Fully-Functional	Eliminated
Cross-platform support	Eliminated	Eliminated
Firebase console linked to test suite(s) with mockups	Eliminated	Fully-Functional

Data inputted will consist of mock data provided by the developers. Blacklists will not be implemented as they are deemed unnecessary for prototype development. Appeals for ratings can be submitted, but will not be managed in the prototype. Deposits for sessions will also be disabled. Meeting confirmation will not be handled due to lack of human resources required to actively check if the tutoring session took place.

4.1 Prototype Architecture

The prototype architecture consists of hardware and software that will be required to meet the minimum requirements defined by the scope of the project. The prototype will be scaled back slightly as the development team will use Old Dominion University as the single test case, leading to differences in database design for the real-world product.

4.1.1 Prototype Hardware

While a majority of the components are held in the cloud, end-users will need to utilize an Android phone to access Tutor Dash. The device must have internet and GPS capabilities. The developers of Tutor Dash will have access to a server hosted by the ODU CS Department to help facilitate development if necessary.

4.1.2 Prototype Software

Tutor dash will utilize a Firebase database server to handle all data required by every component of the prototype. The application will be developed in Android Studio, including the algorithms that leverage external APIs such as Google Calendar, Google Maps, Firebase, and Braintree. For Firebase and Braintree, Android API 16 will be used as the minimum functional requirement. Primary focus will be placed on five algorithms: a PDF transcript parser (which will handle the validation of uploaded transcripts), a relative distance calculator (which will determine whether or not an in-person tutoring session is feasible), a pay-rate calculator (which will factor in course popularity, course demand, and rates when determining a tutor’s suggested base pay), a web-conference creator (which will handle online tutoring sessions through services already provided by Google), and payment logic (which will handle payments based on if the session actually occurred, as well as the length of the session).

4.2 Critical Design Components

As mentioned in Section 4, Tutor Dash will be comprised of three major components: the database, the UI/UX, and the algorithms. The back-end algorithms will work utilize information hosted on the database in order to complete tasks, all of which will be displayed in a manageable user interface.

4.2.1 Database

The Tutor Dash prototype will utilize Firebase, which is a mobile and web application development platform that is primarily hosted on cloud servers provided by Google. It differs from other relational database software in that no tables or rows are utilized; all data is stored in documents found in application-defined collections. User information, school information, schedules, messages, and reviews will all be hosted via Firebase. It is highly scalable, provides easy-to-use APIs, and it syncs data across the board in real-time.

[This space intentionally left blank]

Tutor Dash will have several documents to contain the information provided by users and the information it generates from its back-end algorithms.

Table 2 - Database Schema

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

Typical fields include a student's UID (or university identification number), name, and email. Table 2 lists all of the relevant fields and documents required by Tutor Dash that will be hosted on Firebase.

4.2.2 UI/UX

The user interface will facilitate all end-user interaction through the use of four separate phases and two additional views.

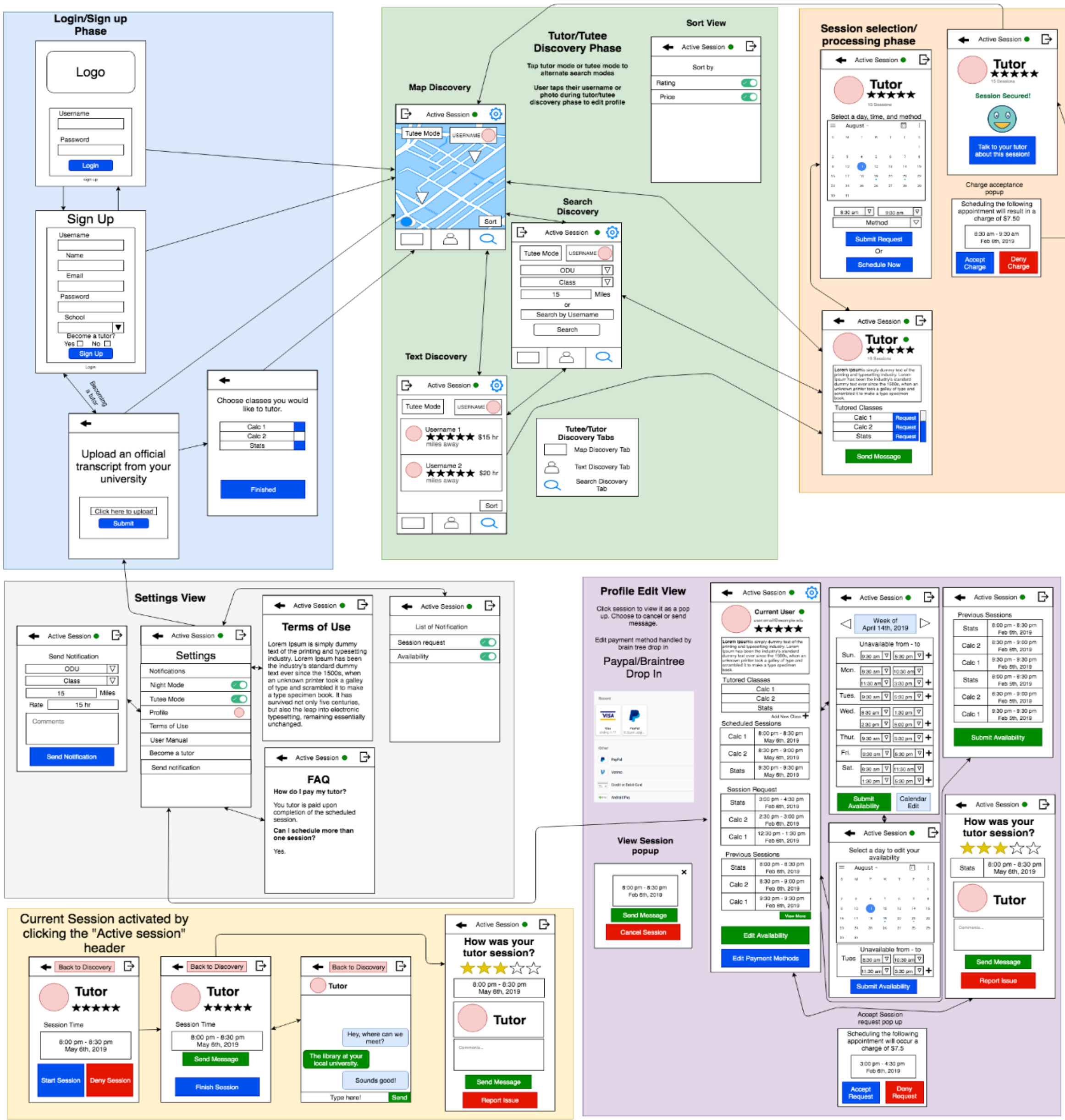

Figure 6 - UI/UX Full Implementation

The four phases include logins and signups, the discovery page for tutors and tutees, the session selector, and the active session.

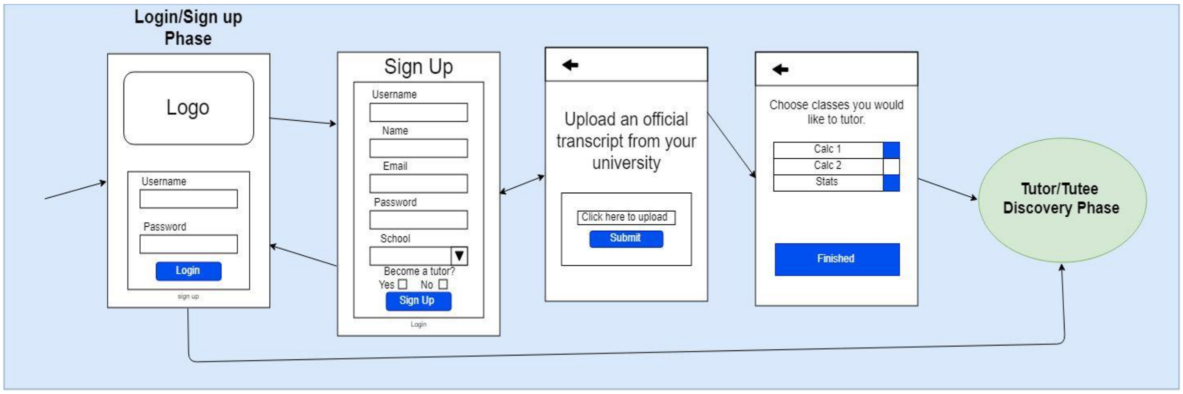

Figure 7 - UI/UX Phase 1

The login phase will allow users to sign in to their accounts or sign up for a new account if they are a new user. Users will be asked to upload a transcript for verification if they want to offer tutoring services. Once the PDF parsing algorithm takes place, tutors will select the courses they wish to provide services for. Upon completion, they will be redirected to the discovery page.

[This space intentionally left blank]

In the discovery phase, users will be presented with a portal that displays tutors seeking out tutees and vice versa.

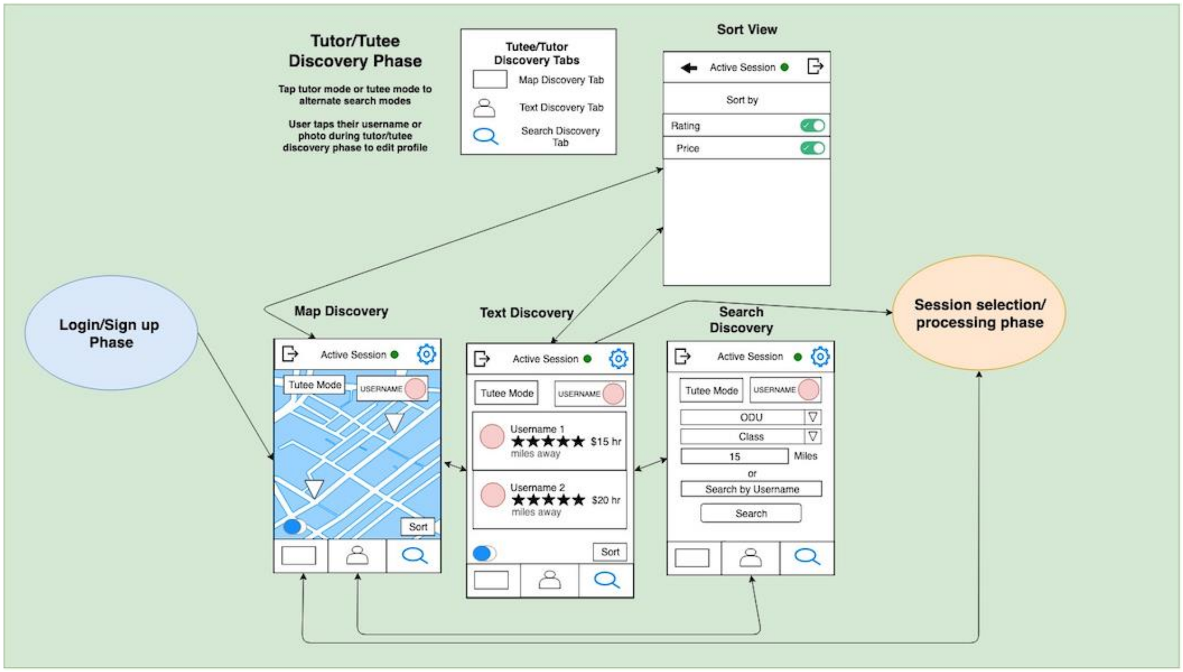

Figure 8 - UI/UX Phase 2

Figure 8 displays the options available to students in greater detail. Users can search using a visual map that tracks location through the Google Maps API. A text discovery option is an alternative option for students that want information in a readable format. For both discovery options, users can narrow results by inputting search criteria.

Aside from discovery, users can directly search for tutors in a separate view. Tutors gain access to a page that allows them to set whether or not they are active, causing them to be filtered from search queries depending on the toggle. If the tutor changes from inactive to active, the appropriate notifications are sent out to alert tutees seeking assistance.

[This space intentionally left blank]

When a session is found and the user makes a selection, they are then taken to the session selection (or processing) phase.

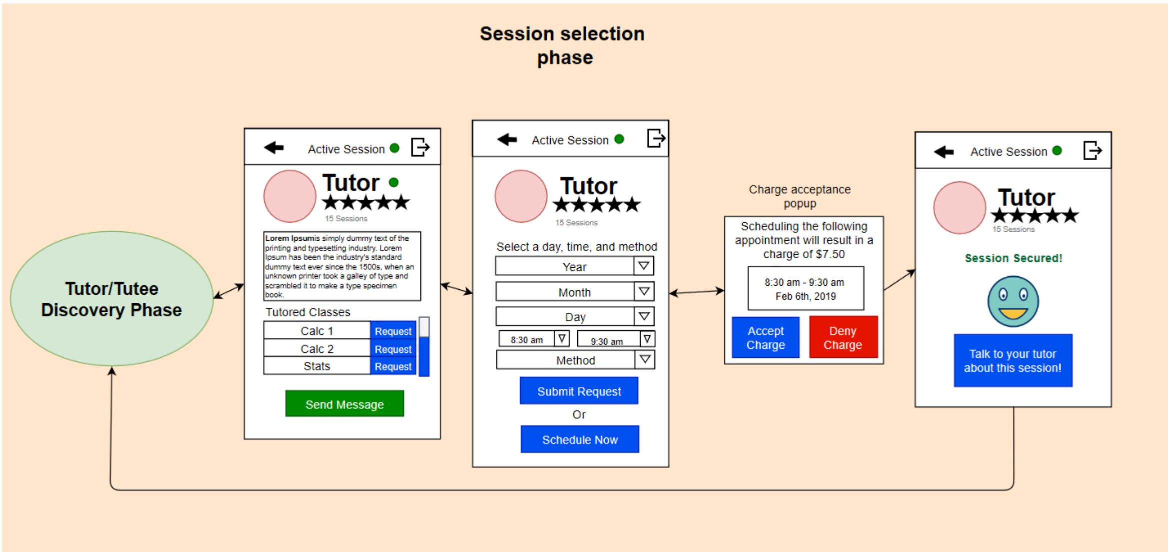

Figure 9 - UI/UX Phase 3

As depicted in Figure 9, users can view information from potential clientele, including their username, rating, biography, and available classes. Tutees can communicate with tutors directly with questions regarding their schedules and other topics. Sessions can be scheduled based on a tutor's available times or as soon as possible. For most phases, currently available tutors are identified by a green dot as shown in Figure 10. The availability of tutors will automatically change based on the schedule they have set, but a manual toggle will be available if users wish to change their status regardless of their scheduled availability.

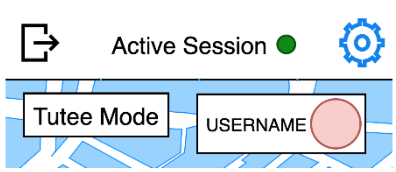

Figure 10 - UI/UX Phase 4A

When a session is scheduled to occur and both the tutor and tutee are present, both parties are moved to the active session phase. Figure 11 outlines the steps both the tutor and tutee will go through when conducting a tutoring session.

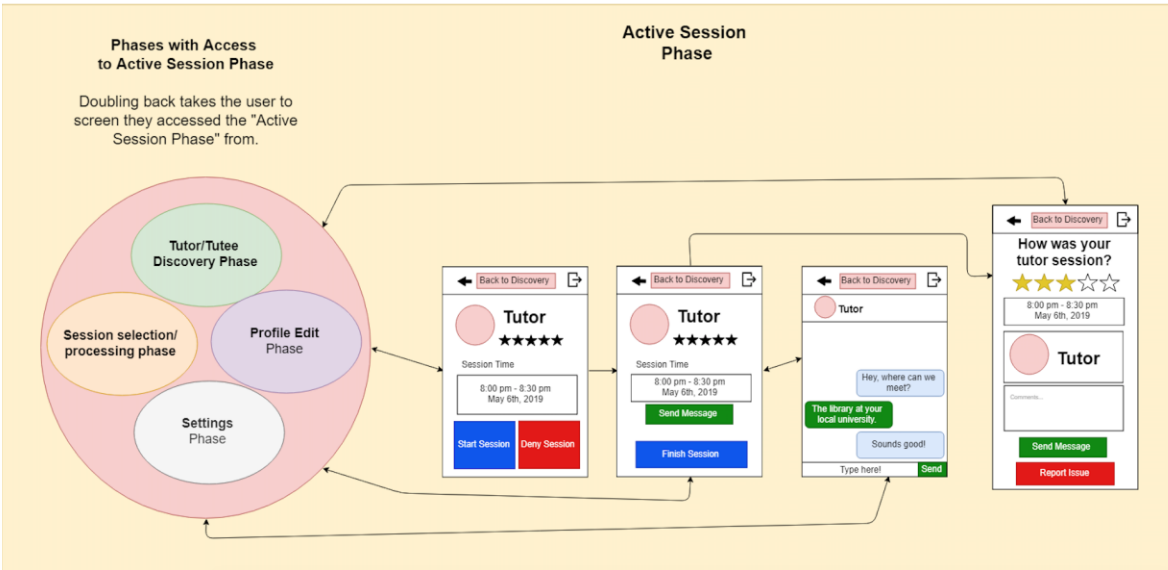

Figure 11 - UI/UX Phase 4B

The initial page will serve as a method to ensure that both parties are present at the time of the meeting. Once the tutoring session has started, a timer will initiate. The session will continue until the timer reaches a certain point or if both parties decide to end the session prematurely. Upon completion of the tutoring session, funds will be sent to the tutor. Both users will be sent to an appropriate page to rate each other out of five stars and will be allowed to report an issue if necessary.

[This space intentionally left blank]

The two additional views include the settings (or options) page and the profile editing page.

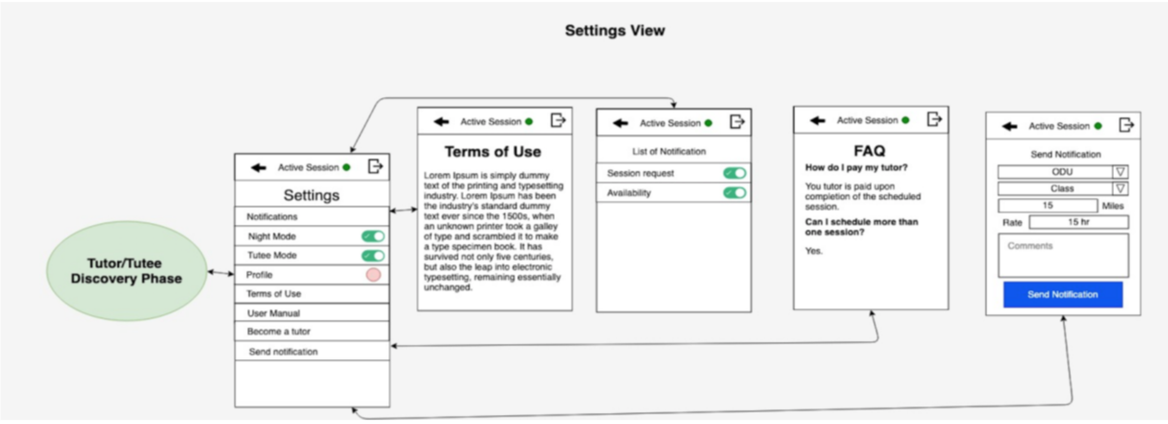

Figure 12 - UI/UX View 1 - Settings

For the former page, users will be able to modify and send out notifications, toggle a night mode (omitted from the prototype), access calendar functionalities, and view their profile. Additional operations include viewing documentation regarding application usage or applying to be a tutor. Figure 12 shows the layout of the view in greater detail.

[This space intentionally left blank]

The latter page will allow users to view all information pertaining to an account in a compact view.

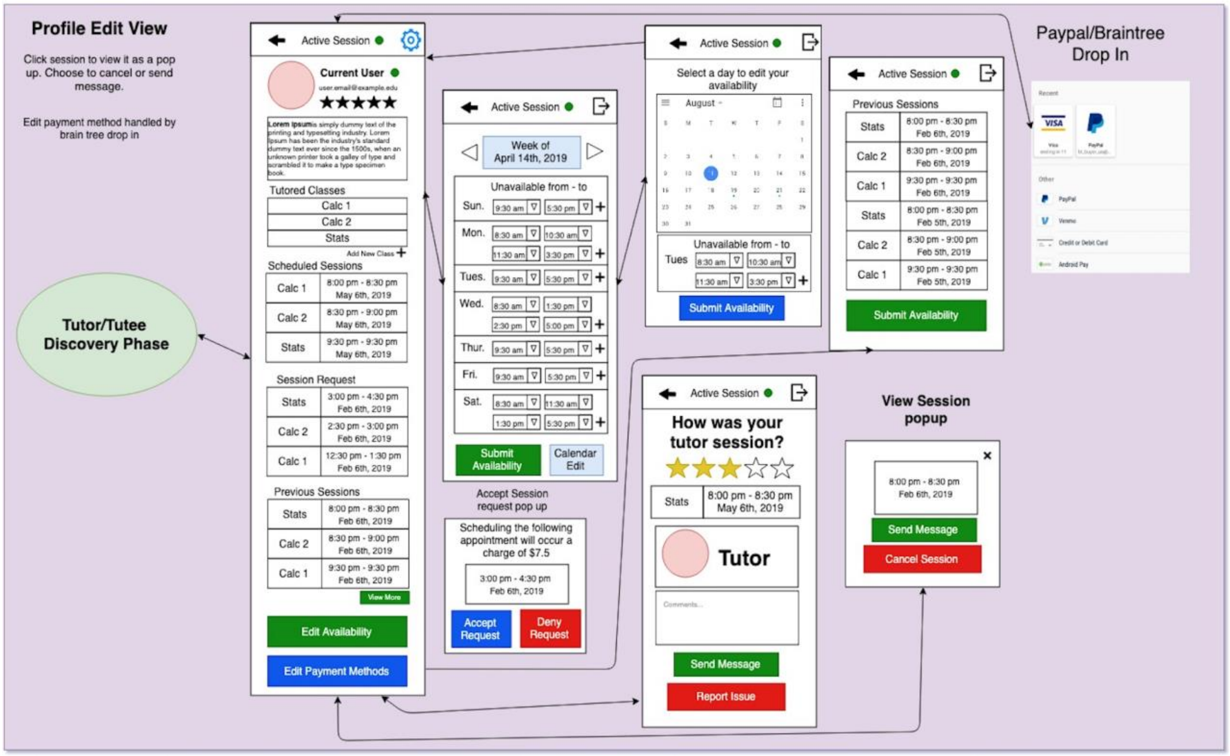

Figure 13 - UI/UX View 2 - User Profile

Account information that can be changed include username, email, biography, tutored classes, scheduled sessions, session requests, previous sessions, and availability. Personal rating cannot be changed. Figure 13 shows the profile edit view in greater detail.

4.2.3 Algorithms

As briefly mentioned in Section 4.1.2, there are five key algorithms that are necessary for the tasks required by Tutor Dash.

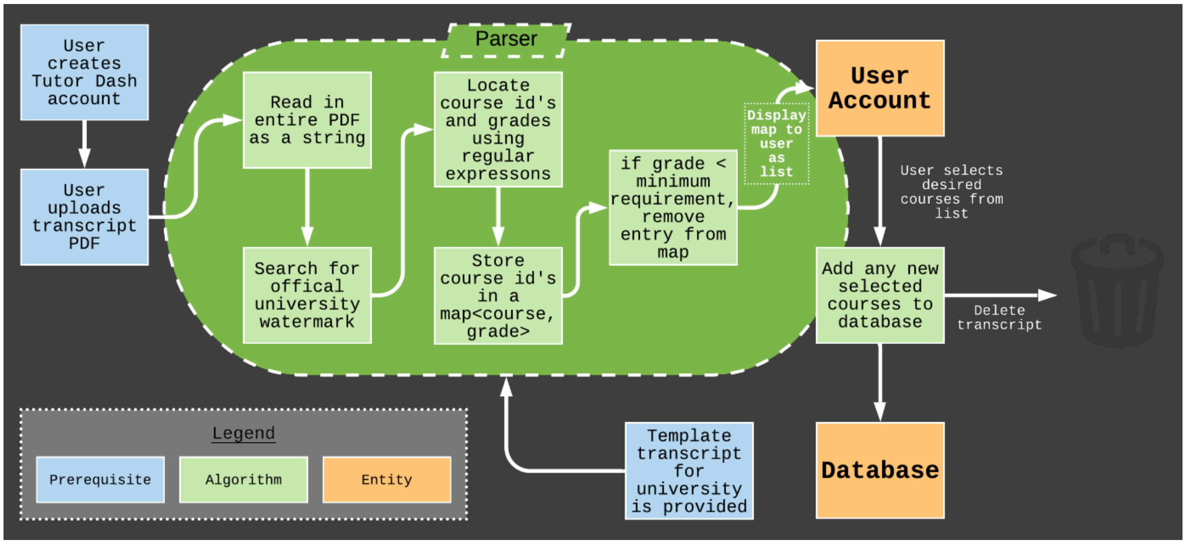

Figure 14 - PDF Transcript Parser Algorithm Flow

Figure 14 illustrates the PDF transcript parser algorithm flowchart. Once a potential tutor creates an account, they are prompted to upload a PDF of their official transcript. The parser will input all the data as a string, search for the university's official watermark, match the course IDs and grades through the university registrar, and store each courses' data in a separate map. Courses that meet the minimum requirements will be displayed to the user. The user will be able to select any of the qualified courses to add to their repertoire, after which the transcript is deleted to prevent personal data from leaking.

[This space intentionally left blank]

The relative distance calculator will provide users with distance information for those seeking tutoring sessions in a traditional, face-to-face environment.

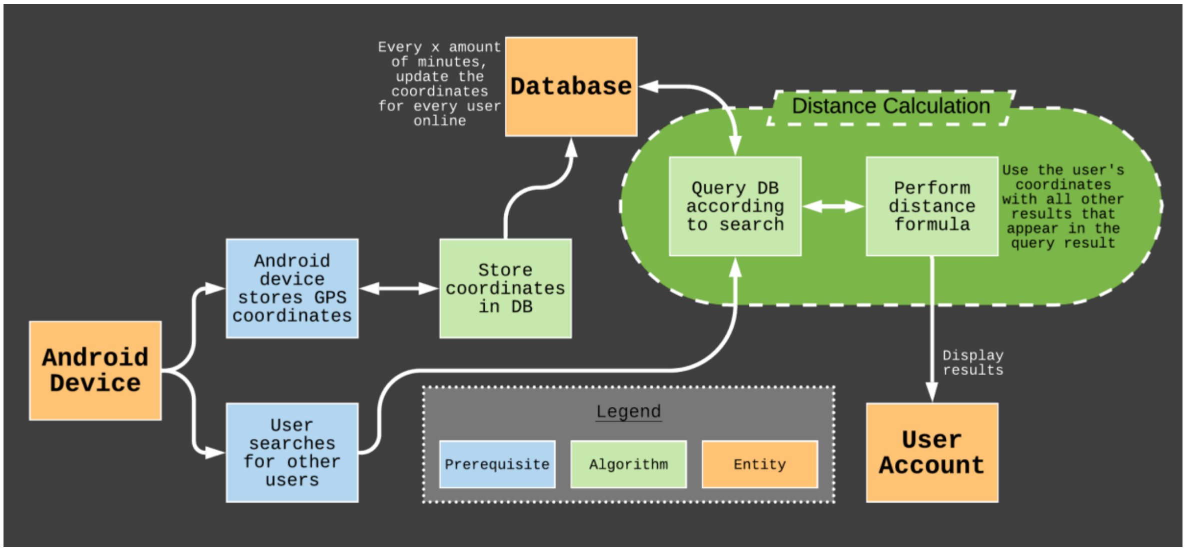

Figure 15 - Relative Distance Calculator Algorithm Flow

Figure 15 illustrates the relative distance calculator flowchart. Utilizing GPS coordinates provided by Android devices, coordinates are stored and updated in the database for online users. A distance formula will be used to determine whether or not an in-person session is feasible for both parties. Acceptable results will be displayed to the user in real-time.

[This space intentionally left blank]

The pay-rates for tutors will be dependent on the competitive pay-rate algorithms, which are designed to help tutors gain more clientele.

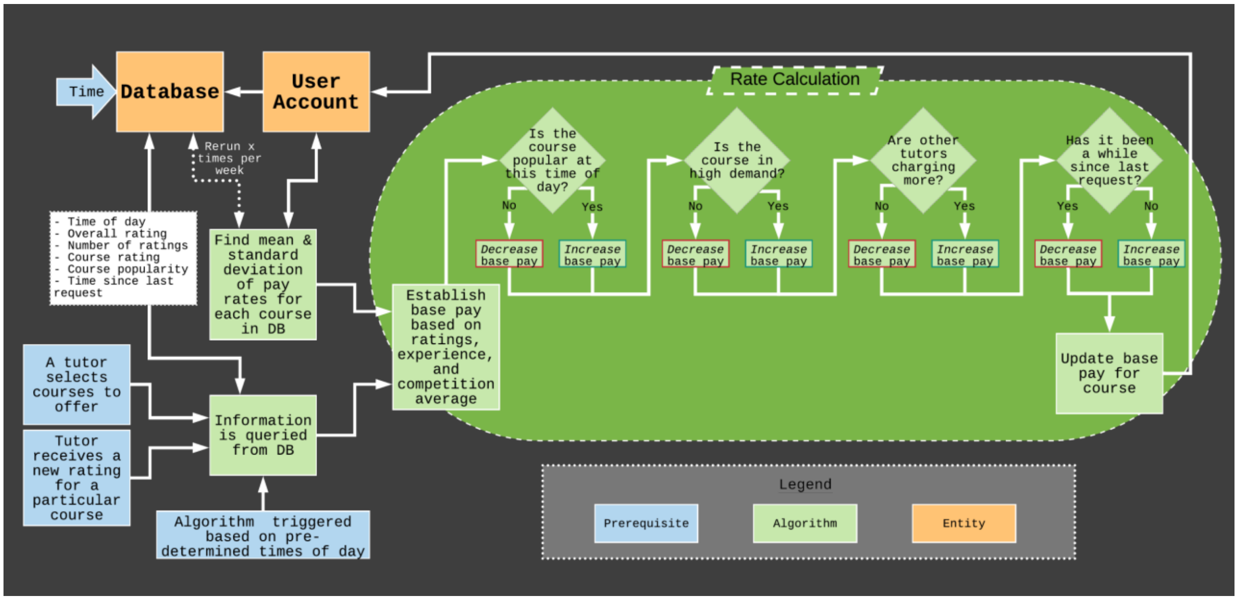

Figure 16 - Competitive Pay-Rate Algorithm Flow

Figure 16 illustrates the flowchart of how Tutor Dash calculates a tutor's base pay. Several parameters dictate how much a tutor will be paid, including the tutor's own ratings (course-specific and overall), the demand and popularity of a course, the amount of experience the tutor has, the time of day, the mean and standard deviation of rates for the course, and the amount of time that has elapsed since the tutor has provided services for the course. Once a fair base pay has been established, the tutors will receive their assigned rate and the databases will update accordingly.

[This space intentionally left blank]

To facilitate online tutoring sessions, a set of algorithms will automatically assign meetings through the use of Google Hangouts.

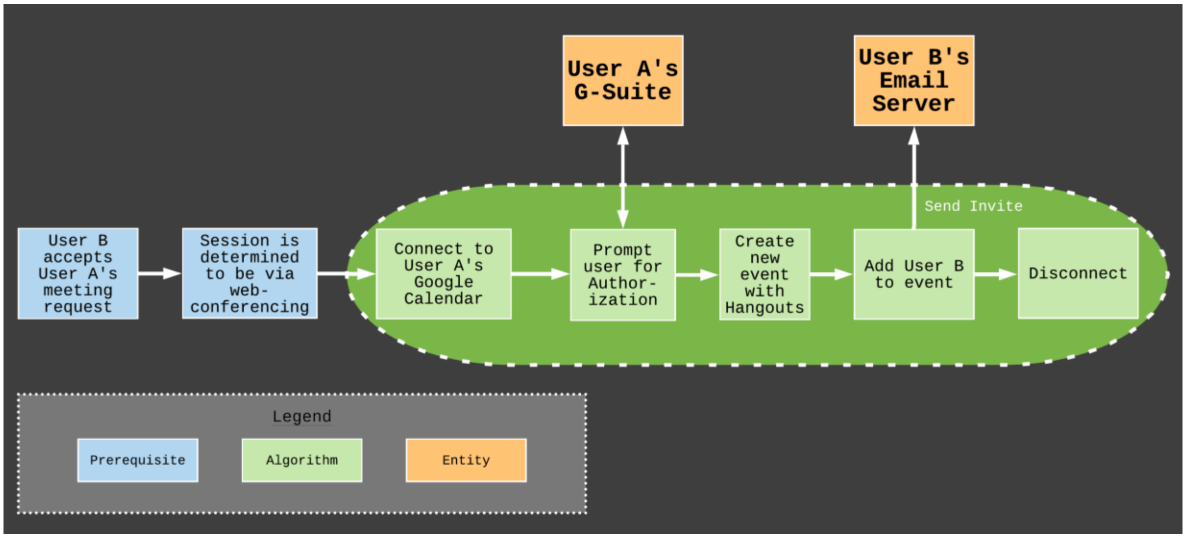

Figure 17 - Web-Conference Appointment Creator Algorithm Flow

The parameters required by this algorithm include the session start time, end time, and the email addresses of both parties. User A (the tutor) will be prompted to authorize Tutor Dash to access their G-Suite. The application will then create an event on Hangouts and proceed to add User B (the tutee) to the event. Figure 17 illustrates the flowchart of the algorithms.

[This space intentionally left blank]

The payment logic algorithms work in tandem with the pay-rate calculator algorithms to handle the distribution of funds from the tutee to the tutor.

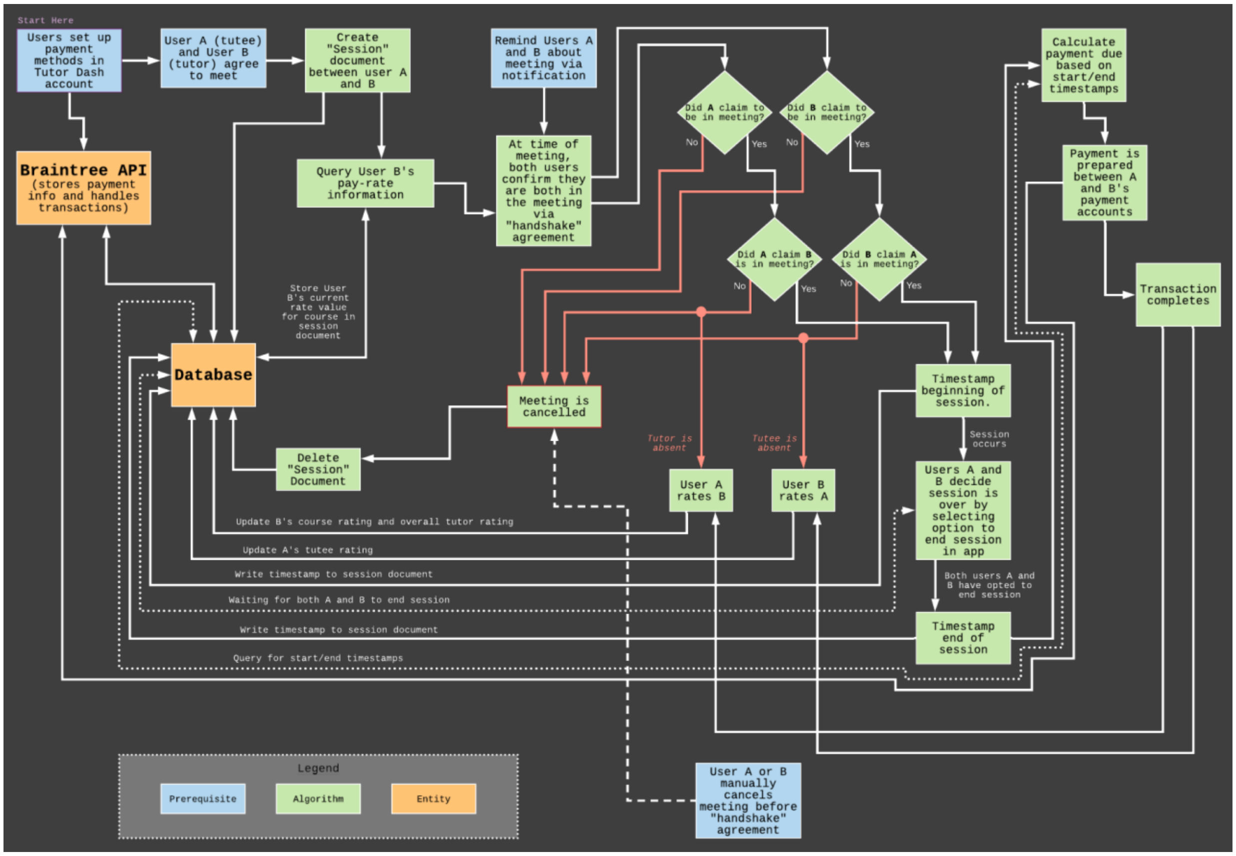

Figure 18 - Payment Logic Algorithm Flow

Upon creating an account, users will need to link a payment method on Tutor Dash to handle transactions. This will be accomplished through the implementation of the Braintree API. A "session" document is created once. Once a session is created, the application retrieves the tutor's pay-rate and begins to go through checkpoints to determine if the tutoring session took place. If all the requirements are met, the application will then prepare how much the tutee owes the tutor based on how long the session took and processes the payments between both accounts through Braintree. Upon completion of the transaction, the users will proceed to rate each other and the appropriate updates are made to the database.

4.3 Prototype Features and Capabilities

Once completed, the prototype will demonstrate an efficient, user-friendly application that connects tutors and tutees with each other as described in Section 2. The prototype will implement a simple payment system, an easy method to upload transcripts, simplified tutoring session scheduling for live meetings or for online meetings hosted on Google Hangouts, a helpful rating system, and a chat system to facilitate communication during the entire process, all of which will be wrapped in an intuitive user interface.

By demonstrating these core functionalities, it will provide an accurate representation of the final application as a whole. The creation of the prototype will effectively solve the problem defined by the case study despite not fulfilling all the requirements of the real-world product listed in table 1. Minimal bugs are expected, but the prototype will undoubtedly remediate the lack of a centralized platform for tutors and tutees to exchange and receiving services.

4.4 Risks and Mitigations

The prototype will address several risks determined by the development team during the prior CS 410 course.

	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

The inclusion of a rating system will minimize user abuse, ensuring that tutees rate fairly by allowing tutors to challenge bogus reviews. It will also mitigate the risk in the event that the tutor or tutee does not have access to hardware that will allow the ability to partake in web conferencing. The PDF transcript parser will ensure that applying tutors are actually qualified for the services they plan to provide. The payment system utilizing the Braintree API will ensure that all funds are handled properly in the event that the tutor or tutee does not meet, meaning payments will only go through once the tutoring session has been confirmed to take place by both parties. The Google Calendar API will work hand in hand with the payment system to ensure that the meetings do not overlap with other sessions. All defined risks, including those that would be typically considered miscellaneous (i.e., violations to the FERPA or illegal usage of the application) are mitigated through methods that are properly defined in Figure 20.

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

Figure 20 - Risk Matrix Descriptions

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
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 (Cont.)

4.5 Prototype Objectives and Goals

Upon completion of the Tutor Dash prototype, the overall objective is to implement all the core functionalities of the application as defined in Section 4.3. A clean user interface will allow navigation within the application to be smooth and free of any hassles. The tutor and tutee matching algorithm will be streamlined to minimize time that would be wasted otherwise. The simple payment system will ensure that tutors are paid fairly with algorithms adjusting payment based on factors including course popularity, course demand, and existing rates provided by other tutors. A proper connection to an outside medium will ensure the quality of tutoring sessions held online. Lastly, the rating system will allow tutees to effectively gauge tutors based on prior sessions, making sure each rating will be indicative of the quality of the experience itself.

4.6 Prototype Development Challenges

The Tutor Dash development team will face several development challenges. One of the major hurdles is the lack of Android application development experience amongst most of the members, which will dictate the sophistication and completion of the prototype itself. Implementing external APIs (such as G Suites, Firebase, and the Braintree payment API) will also be challenging for members not used to handling them.

Additional challenges include implementing a proper front-end user interface with back-end components that include the algorithms and databases required for fulfilling the prototype's objectives. The members will need to develop an efficient medium that handles communication and payments between tutees and tutors, making sure all prerequisites are met in order for a successful tutoring session to occur. For communication, algorithms need to be developed so that notifications for matching tutors and tutees work properly, in addition to a messaging system that will help with any additional requirements set in place by either party. For payments, the algorithm that calculates rates will be difficult to implement as it needs to take into account several factors defined in Section 4.2.3. The group will need to provide simulated data in order to properly test this algorithm. To finalize the entire process, a "handshake" agreement will need to be implemented so that the tutoring session is confirmed, ensuring that both parties are aware of when and how the meeting will take place.

All development factors are limited by a time constraint imposed by the nature of the course, which may result in some prototype features being omitted or incomplete upon the end of the semester. The team will need to complete tasks in a timely matter while testing the overall user experience to ensure a quality prototype is delivered.

Glossary

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

DFWI: An acronym for Drop/Fail/Withdraw/Incomplete. This relates to university course incompletion status.

DFWI 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.

Indirect Competitor: Another product or company which is involved in the same domain space as Tutor Dash, but does 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.

References

1. "Academic Tutoring in Comprehensive Universities." Hanover Research, 2014. URL: https://www.hanoverresearch.com/wp-content/uploads/2017/08/Academic-Tutoring-in-Comprehensive-Universities.pdf
2. "Academics." Old Dominion University, 8 Jan. 2019. URL: www.odu.edu/academics
3. "Campus Tutoring." Old Dominion University, 19 Jan. 2019. URL: www.odu.edu/success/academic/tutoring#tab125=0
4. "Course-Specific Tutoring." Old Dominion University, 19 Jan. 2019. URL: www.odu.edu/success/academic/tutoring/course-specific
5. "Courses of Instruction." Old Dominion University, Feb. 2019. URL: catalog.odu.edu/courses/
6. 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
7. 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=Email_to_authors_&utm_medium=Email&utm_content=T1_11.5e1_author&utm_campaign=Email_publication&field=&journalName=Frontiers_in_Human_Neuroscience&id=239492
8. "Facebook - Groups." Facebook Help Center, Facebook, 2019. URL: www.facebook.com/help/1629740080681586?helpref=hc_global_nav
9. "Find a Local In-Home Tutor Today." HeyTutor, HeyTutor LLC. URL: heytutor.com/
10. 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
11. "Peer Assisted Learning" BMC Education, 8 March 2006 URL: https://bmcmededuc.biomedcentral.com/articles/10.1186/1472-6920-6-18
12. 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=JnNpdGU9ZWhvc3QtbGl2ZSZzY29wZT1zaXRl#AN=114789419&db=ehh
13. 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

14. "Skooli Tutors Online." Skooli Online Tutoring, Skooli, Feb. 2019. URL: www.skooli.com/prices/students
15. "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
16. Topping, J. Keith. "Trends in Peer Learning", 19 Jan 2007 URL: https://www.tandfonline.com/doi/full/10.1080/01443410500345172?scroll=top&needAccess=tru
17. "Tutor Matching Service - How It Works." Tutor Matching Service, Tutor Matching Service, 2019. URL: tutormatchingservice.com/#/about
18. "Tutor.com - The Princeton Review." Tutor.com, The Princeton Review, URL: www.tutor.com
19. "Tutors - Care.com." Care.com, Care.com, Feb. 2019. URL: www.care.com/tutors
20. "Wyzant." Wyzant Resources, Wyzant Inc., Feb. 2019. URL: www.wyzant.com/howitworks/students