Lab 2 – Tutor Dash Product Specification

Edwin R. Ordona

Old Dominion University

CS 411W

Professor T. Kennedy

November 8, 2019

Version 2

Table of Contents

List of Tables

List of Figures

Lab 2 – Tutor Dash Product Specification

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 [*Peer Assisted Learning, 2006*]. The availability of these sessions, however, can sometimes be unpredictable without browsing through several resources provided by campuses or third-party services. Figure 1 illustrates how tutees are reliant on the availability, location and skillsets of tutors.

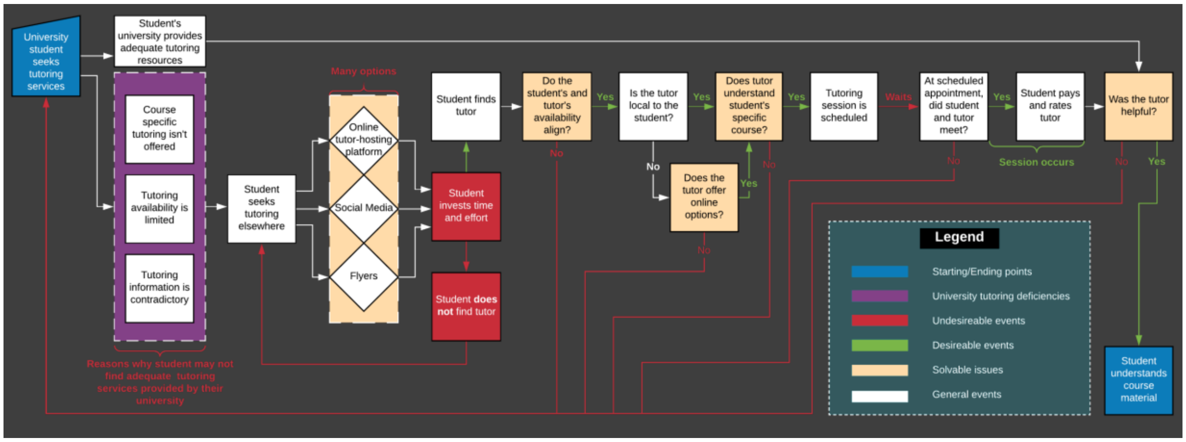
University student seeks tutoring services
Student's university provides adequate tutoring resources
Course specific tutoring isn't offered
Tutoring availability is limited
Tutoring information is contradictory
Reasons why student may not find adequate tutoring services provided by their university
Student seeks tutoring elsewhere
Many options
Online tutor-hosting platform
Social Media
Flyers
Student finds tutor
Student invests time and effort
Student does not find tutor
Do the student's and tutor's availability align?
Yes
No
Is the tutor local to the student?
Yes
No
Does the tutor offer online options?
Yes
No
Does tutor understand student's specific course?
Yes
No
Tutoring session is scheduled
Waits
At scheduled appointment, did student and tutor meet?
Yes
No
Student pays and rates tutor
Session occurs
Was the tutor helpful?
No
Yes
Student understands course material
Legend
🟦 Starting/Ending points
🟪 University tutoring deficiencies
🟥 Undesireable events
🟩 Desireable events
🟧 Solvable issues
⬜ General events

Figure 1 - Students Seeking Academic Assistance

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 illustrates the overall process and the various obstacles that students looking to offer tutoring services will have to go through.

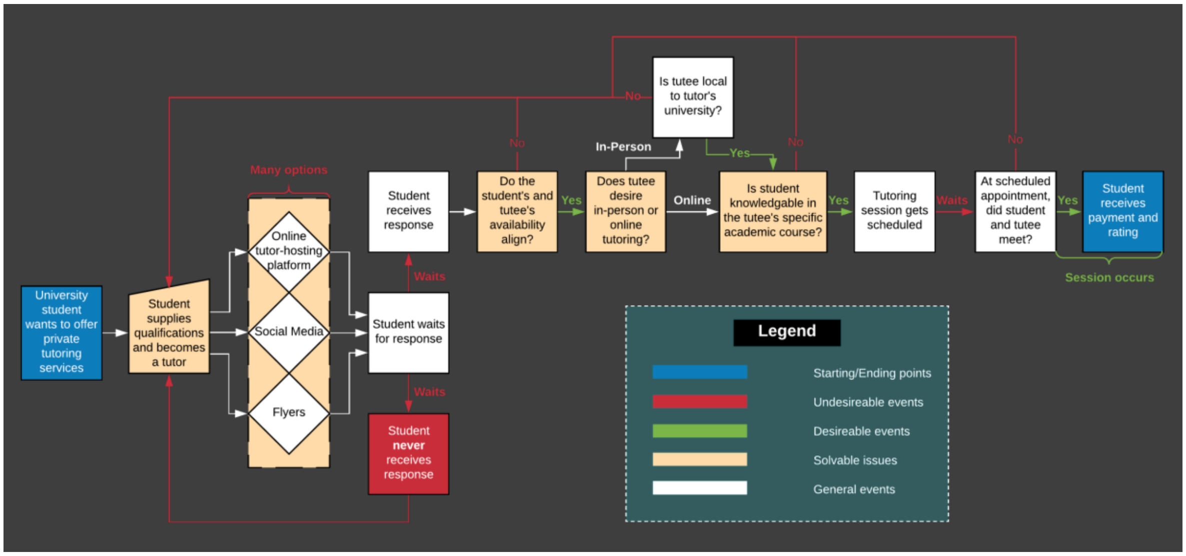
University student wants to offer private tutoring services
Student supplies qualifications and becomes a tutor
Many options
Online tutor-hosting platform
Social Media
Flyers
Student waits for response
Waits
Student receives response
Waits
Student never receives response
Do the student's and tutee's availability align?
No
Yes
Does tutee desire in-person or online tutoring?
In-Person
Online
Is tutee local to tutor's university?
No
Yes
Is student knowledgeable in the tutee's specific academic course?
No
Yes
Tutoring session gets scheduled
Waits
At scheduled appointment, did student and tutee meet?
No
Yes
Student receives payment and rating
Session occurs
Legend
Starting/Ending points
Undesireable events
Desireable events
Solvable issues
General events

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.

1.1 Purpose

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. The application will not introduce an official tutoring resource, nor will it interfere with existing resources. The overall purpose of Tutor Dash is to provide university students a centralized platform that connects tutors and tutees directly to each other, essentially acting as a supplement for existing tutoring resources provided by campuses and services provided 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.

1.2 Scope

Tutor Dash has several objectives that will be met upon completion of its core functionalities. These algorithms will work in tandem to ensure a quality user experience.

The transcript parser will help potential tutors verify their qualified courses automatically, with little to no user interaction required. The relative distance calculator algorithm will help tutees find nearby tutors that are available. The competitive pay-rate algorithm and the automatic payment logic algorithm will ensure that tutors are paid fairly in respect to their offered courses and that money between both tutors and tutees are handled securely by leveraging Braintree's payment APIs. For the online tutoring sessions, the web-conference appointment creator will utilize Hangouts to create online appointments between both parties automatically, and it will implement a handshake agreement to ensure the session takes place with both parties present. A chat messaging system will also be available to facilitate communication prior to the actual session. Afterwards, a rating system will gauge the performance of both parties so that tutors are encouraged to provide a quality experience to all of their future clients, and tutees are encouraged to show up to their assigned sessions and act appropriate.

To summarize, the Tutor Dash prototype will be an accurate representation of the real-world product with a few optional quality-of-life features being omitted from development. The initial case study that highlights the problem of tutors lacking a platform to extend their services' reach and tutees lacking an efficient way to search for available tutors at any given time will be solved upon completion of the prototype.

1.3 Definitions, Acronyms, and Abbreviations

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.

Serverless Architecture: Concerning database interactions over a network server, this type of architecture implies that the server's implementation is invisible (or abstracted) to the team developing the product using the actual server.

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.

1.4 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=Email_to_authors_&utm_medium=Email&utm_content=T1_11.5e1_author&utm_campaign=Email_publication&field=&journalName=Frontiers_in_Human_Neuroscience&id=239492

"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

“Lab 1 - Tutor Dash Product Description.” Team Gold. 26 Sept. 2019. URL: https://www.cs.odu.edu/~411gold

"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=JnNpdGU9ZWhvc3QtbGl2ZSZzY29wZT1zaXRl#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/10.1186/1472-6920-14-115

Topping, J. Keith. "Trends in Peer Learning", 19 Jan 2007 URL: https://www.tandfonline.com/doi/full/10.1080/01443410500345172?scroll=top&needAccess=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

1.5 Overview

The Tutor Dash product specification will outline the various software configurations, interfaces, and features of the prototype. The majority of the software utilized will be hosted online, with users needing either an Android phone or emulator as the sole user interface. Section 3 of the product specification is provided in a separate document and will list all of the specific requirements needed by the prototype.

2. General Description

Tutor Dash will be an Android application that users will need to download onto their devices. The target audience will be university students who want to provide or receive tutoring services. As such, all users will require an official email provided by their university. The prototype will focus on the completion of the core capabilities mentioned in Section 1.2. Tutors will be able to upload their transcripts, advertise their services, and receive fair compensation. Tutees will be able to search for tutors and their desired courses in real-time via a map or text discovery page. The handling of the session creation will be semi-autonomous, requiring both parties to be present to ensure a successful meeting. Table 1 lists all the features that are shared between the prototype and the final product, in addition to the features that are omitted or partially implemented that differentiate the two versions.

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
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

2.1 Prototype Architecture Description

Since the front and back-end of Tutor Dash is developed primarily in Android Studio, most of the functionalities will be accessed through an Android device. Figure 3 shows the major functional components required by Tutor Dash in order to operate as intended.

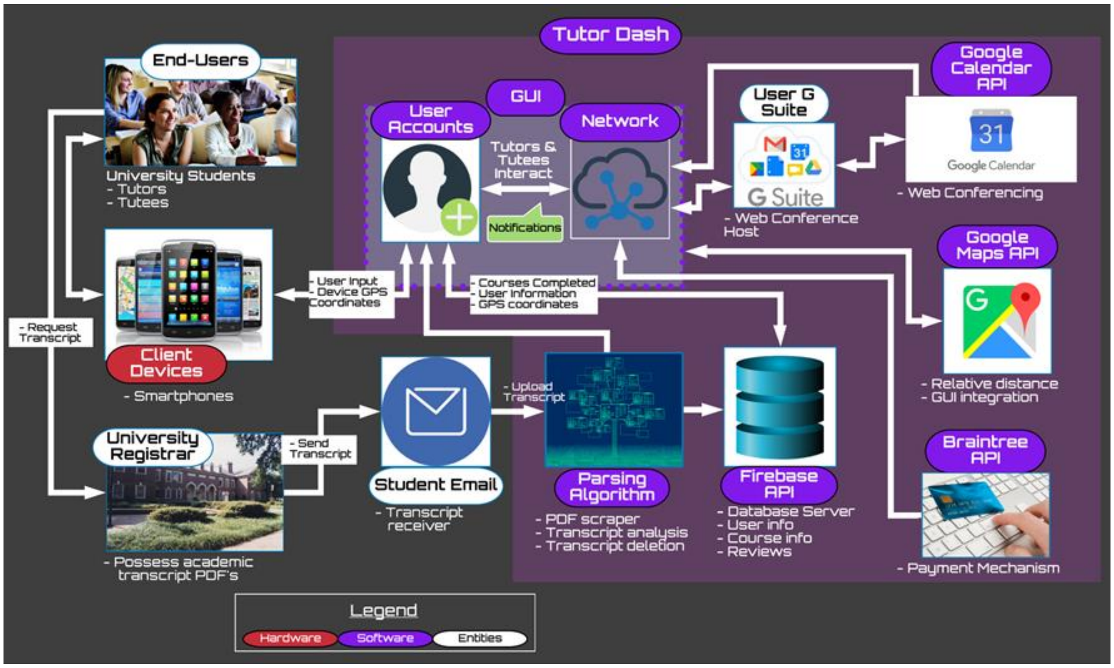

Figure 3 - Major Functional Component Diagram

A number of APIs are leveraged for the core functionalities of Tutor Dash. Braintree is used for secure payment handling between tutors and tutees. Google Maps is used for GUI integration and location data that will be uploaded to a database. G-Suite and Google Calendar will be used to handle the creation of web conferences for online tutoring sessions. Firebase is used for all the database operations that will be used throughout the entirety of the application.

All other components are designed by the development team of Tutor Dash. Users will have their own accounts tied to the email address they signed up with that contain their information, qualified courses, coordinates, and payment methods. The parsing algorithm will be written to accept official transcripts provided by the university. The payment, distance, and conference scheduling code will be written to utilize the APIs mentioned above.

The entirety of Tutor Dash will be displayed as a simple front-end interface that guides users throughout its features. Figure 4 illustrates the initial pages new and returning users will see upon selecting the application on their devices.

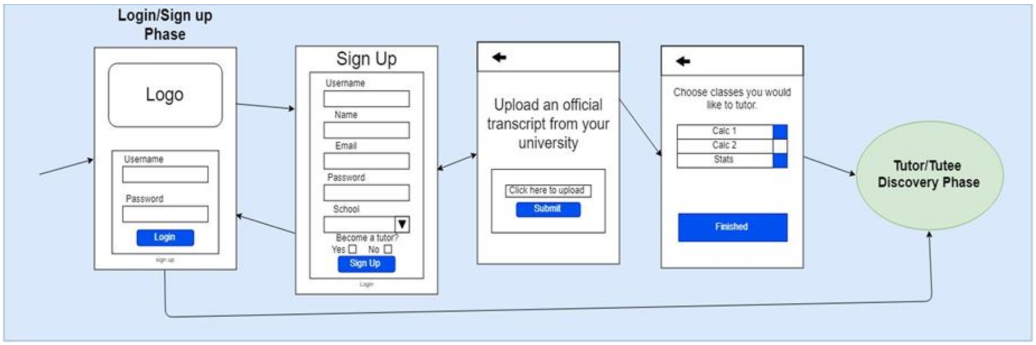

Figure 4 - UI/UX Phase 1

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 main discovery pages detailed in Figure 5.

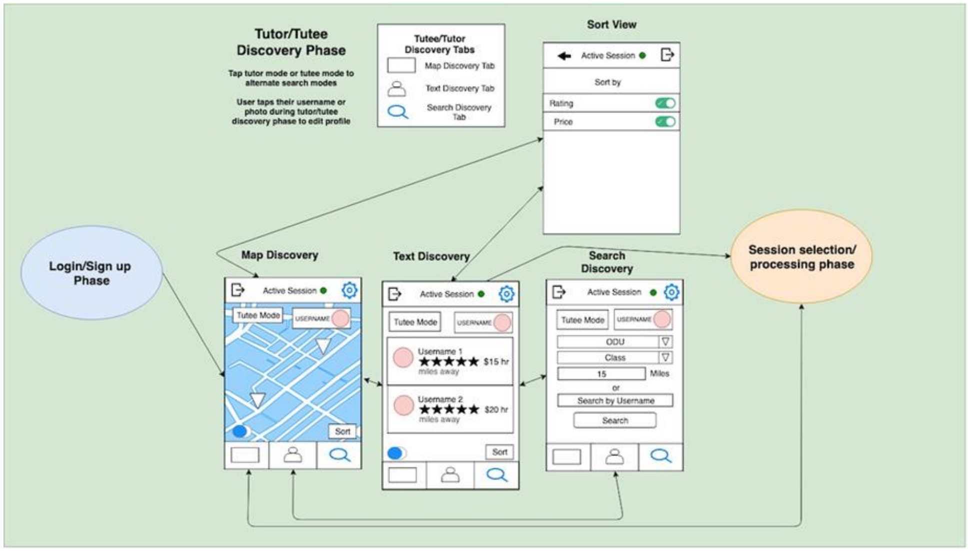

Figure 5 - UI/UX Phase 2

After the first phase, users will be sent initially to the map discovery page for a visual representation of their location utilizing Google Maps. From there users have several navigational options. On the top, users will be able to sign out, access their profile, and see the name of the page they are currently on. On the bottom, a navigational bar will allow users to switch between the map discovery page, the text discovery page, and the search discovery page. The text discovery page shows available tutors in a text format, listing their rating, username and picture. The search discovery page allows users to search for specific courses or tutors. Upon submitting a search query, the text and map discovery pages will update accordingly to filter results based on user input. The text and map discovery pages will have a sort button to order results based on rating, distance, and price. All three pages will have the option to switch between tutor and tutee mode, as well as an availability toggle.

When a tutee selects a tutor, they will be presented with options to view additional details about the tutor and request a session. Figure 6 outlines the pages tutors and tutees will access when creating or processing a tutoring session.

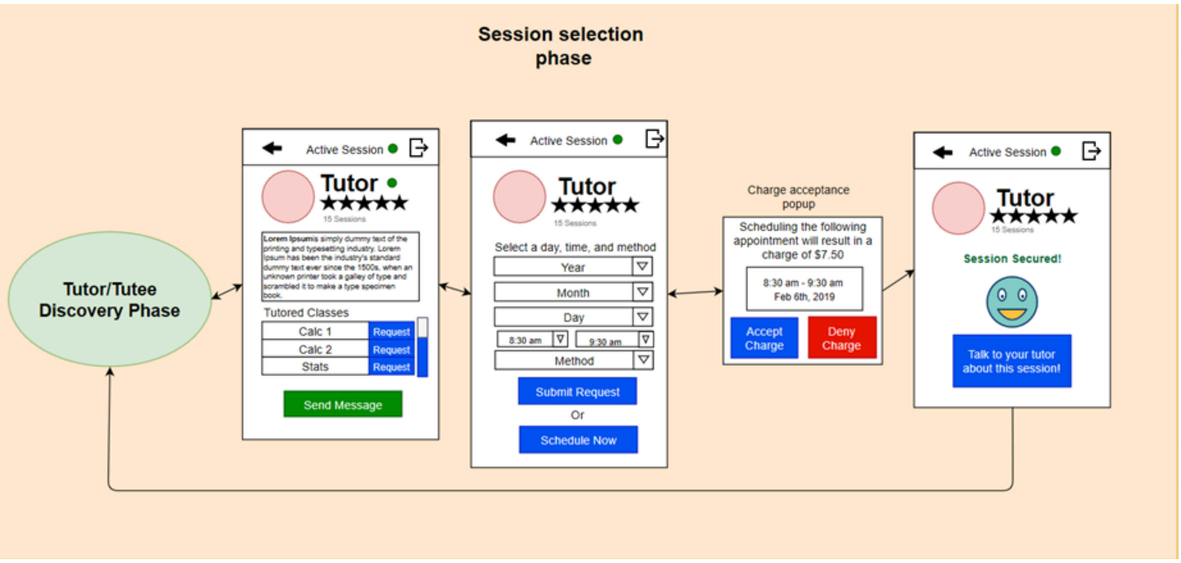
Session selection phase
Tutor/Tutee Discovery Phase
Active Session
Tutor
★★★★★
15 Sessions
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.
Tutored Classes
Calc 1
Request
Calc 2
Request
Stats
Request
Send Message
Active Session
Tutor
★★★★★
15 Sessions
Select a day, time, and method
Year
Month
Day
8:30 am
9:30 am
Method
Submit Request
Or
Schedule Now
Charge acceptance popup
Scheduling the following appointment will result in a charge of $7.50
8:30 am - 9:30 am
Feb 6th, 2019
Accept Charge
Deny Charge
Active Session
Tutor
★★★★★
15 Sessions
Session Secured!
Talk to your tutor about this session!

Figure 6 - UI/UX Phase 3

On the first page, tutees can request a specific class or send a message to the tutor asking for additional details. Requesting a session will allow tutees to select when the session will occur and the method of instruction. Once a request has been submitted and the tutee accepts the potential charge, the tutor is prompted to either accept or deny the session request. If the tutor accepts, the session starts at the predetermined time, at which the users will be sent to the active session page defined in Figure 7.

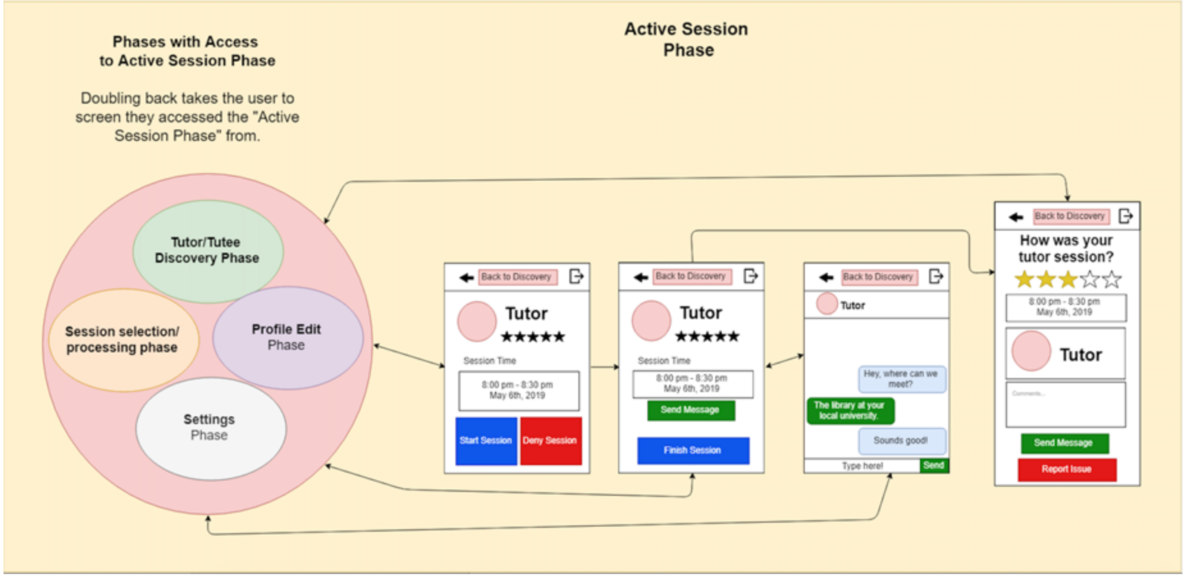

Figure 7 - UI/UX Phase 4B

As the session occurs, the users will be presented with buttons to send a message, finish the current session, or return to the discovery pages to look for a new session. Sending a message will show a chat box between the tutor and tutee. Finishing a session will prompt users to rate each other, after which both users will be sent back to the discovery pages.

[This space intentionally left blank]

Figure 8 shows the settings available to users and can be accessed by the cogwheel in the upper right corner of the discovery pages.

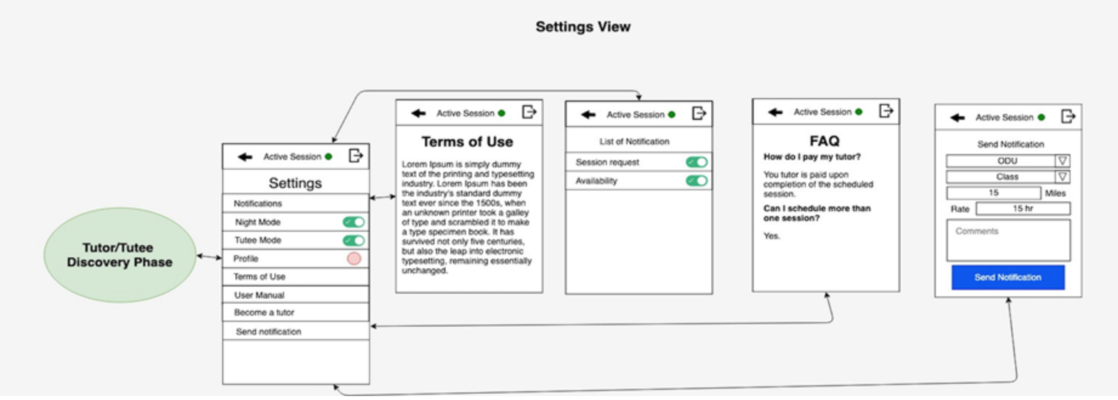
Settings View
Tutor/Tutee Discovery Phase
Active Session
Settings
Notifications
Night Mode
Tutee Mode
Profile
Terms of Use
User Manual
Become a tutor
Send notification
Terms of Use
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.
List of Notification
Session request
Availability
FAQ
How do I pay my tutor?
You tutor is paid upon completion of the scheduled session.
Can I schedule more than one session?
Yes.
Send Notification
ODU
Class
15
Miles
Rate
15 hr
Comments
Send Notification

Figure 8 - UI/UX View 1 – Settings

Aside from night mode (which will be omitted from the prototype), users will be able to toggle between tutor and tutee mode, edit their profile (depicted in Figure 9), and view the user manual and other related pages. Users can send out notifications if they are available for tutoring or if they are searching for tutoring services.

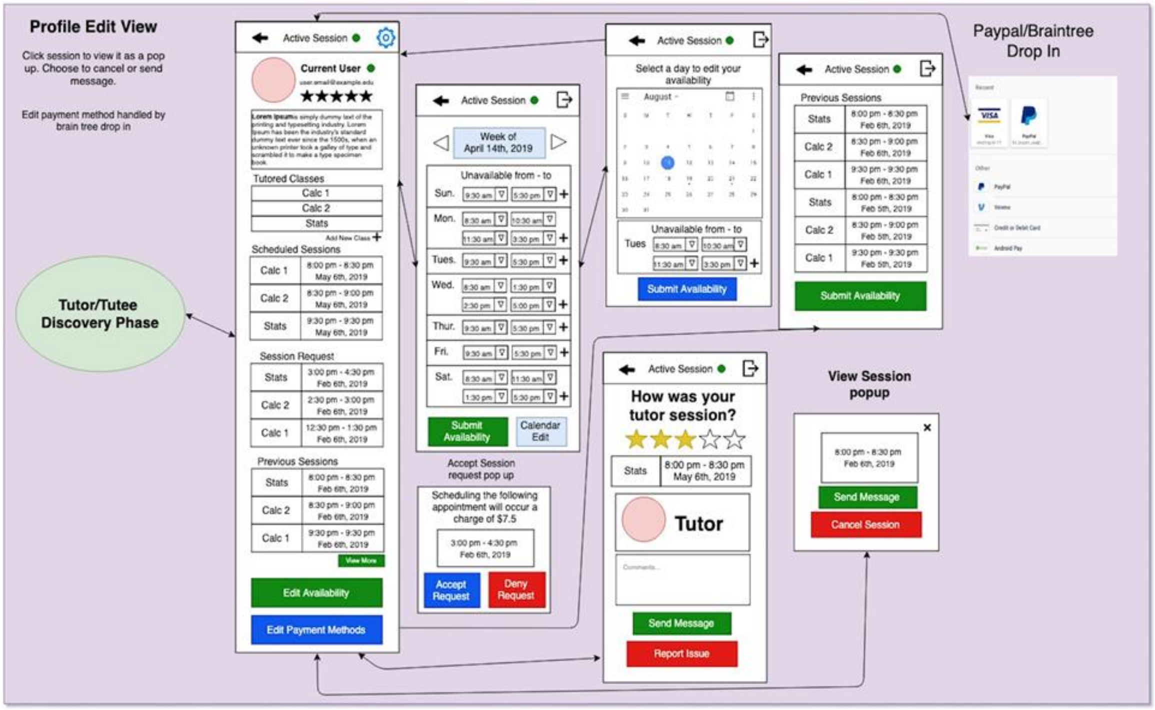
Profile Edit View
Click session to view it as a pop up. Choose to cancel or send message.
Edit payment method handled by brain tree drop in
Tutor/Tutee Discovery Phase
Active Session
Current User
user.email@example.edu
★★★★★
Tutored Classes
Calc 1
Calc 2
Stats
Add New Class +
Scheduled Sessions
Calc 1 8:00 pm - 8:30 pm May 6th, 2019
Calc 2 8:30 pm - 9:00 pm May 6th, 2019
Stats 9:30 pm - 9:30 pm May 6th, 2019
Session Request
Stats 3:00 pm - 4:30 pm Feb 6th, 2019
Calc 2 2:30 pm - 3:00 pm Feb 6th, 2019
Calc 1 12:30 pm - 1:30 pm Feb 6th, 2019
Previous Sessions
Stats 8:00 pm - 8:30 pm Feb 6th, 2019
Calc 2 8:30 pm - 9:00 pm Feb 6th, 2019
Calc 1 9:30 pm - 9:30 pm Feb 6th, 2019
View More
Edit Availability
Edit Payment Methods
Week of April 14th, 2019
Unavailable from - to
Sun. Mon. Tues. Wed. Thur. Fri. Sat.
Submit Availability
Calendar Edit
Accept Session request pop up
Scheduling the following appointment will occur a charge of $7.5
3:00 pm - 4:30 pm Feb 6th, 2019
Accept Request
Deny Request
Select a day to edit your availability
August
Unavailable from - to
Tues
Submit Availability
Previous Sessions
Stats 8:00 pm - 8:30 pm Feb 6th, 2019
Calc 2 8:30 pm - 9:00 pm Feb 6th, 2019
Calc 1 9:30 pm - 9:30 pm Feb 6th, 2019
Stats 8:00 pm - 8:30 pm Feb 5th, 2019
Calc 2 8:30 pm - 9:00 pm Feb 5th, 2019
Calc 1 9:30 pm - 9:30 pm Feb 5th, 2019
Submit Availability
Paypal/Braintree Drop In
How was your tutor session?
Stats 8:00 pm - 8:30 pm May 6th, 2019
Tutor
Comments...
Send Message
Report Issue
View Session popup
8:00 pm - 8:30 pm Feb 6th, 2019
Send Message
Cancel Session

Figure 9 - UI/UX View 2 - User Profile

The profile edit view allows users to see their scheduled sessions, session requests that have been sent or received, and previously completed sessions. Users are able to modify and submit their availability which will automatically handle the toggle mentioned in Figure 5. Payment options are also handled in the appropriate page.

2.2 Prototype Functional Description

Tutor Dash will provide five main functionalities that were summarized in Sections 1.2 and 2.1. More details are described in this section, with specific requirements listed in Section 3.

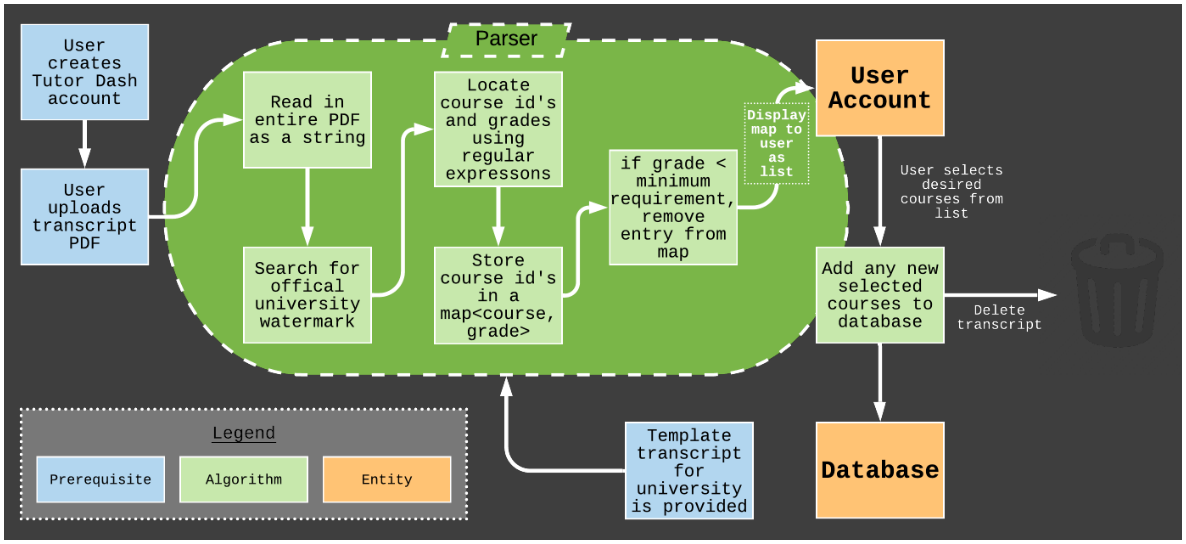
User creates Tutor Dash account
User uploads transcript PDF
Parser
Read in entire PDF as a string
Search for offical university watermark
Locate course id's and grades using regular expressions
Store course id's in a map<course, grade>
if grade < minimum requirement, remove entry from map
Display map to user as list
User Account
User selects desired courses from list
Add any new selected courses to database
Delete transcript
Database
Template transcript for university is provided
Legend
Prerequisite
Algorithm
Entity

Figure 10 - PDF Transcript Parser Algorithm Flow

Figure 10 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.

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

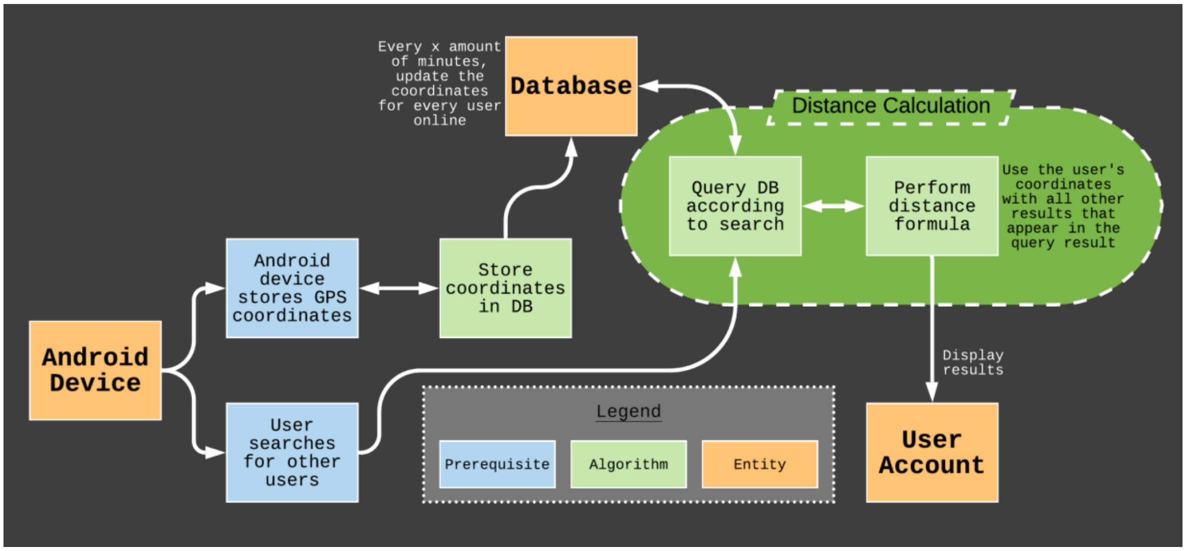
Every x amount of minutes, update the coordinates for every user online
Database
Distance Calculation
Query DB according to search
Perform distance formula
Use the user's coordinates with all other results that appear in the query result
Android device stores GPS coordinates
Store coordinates in DB
Android Device
Display results
User searches for other users
Legend
Prerequisite
Algorithm
Entity
User Account

Figure 11 - Relative Distance Calculator Algorithm Flow

Figure 11 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.

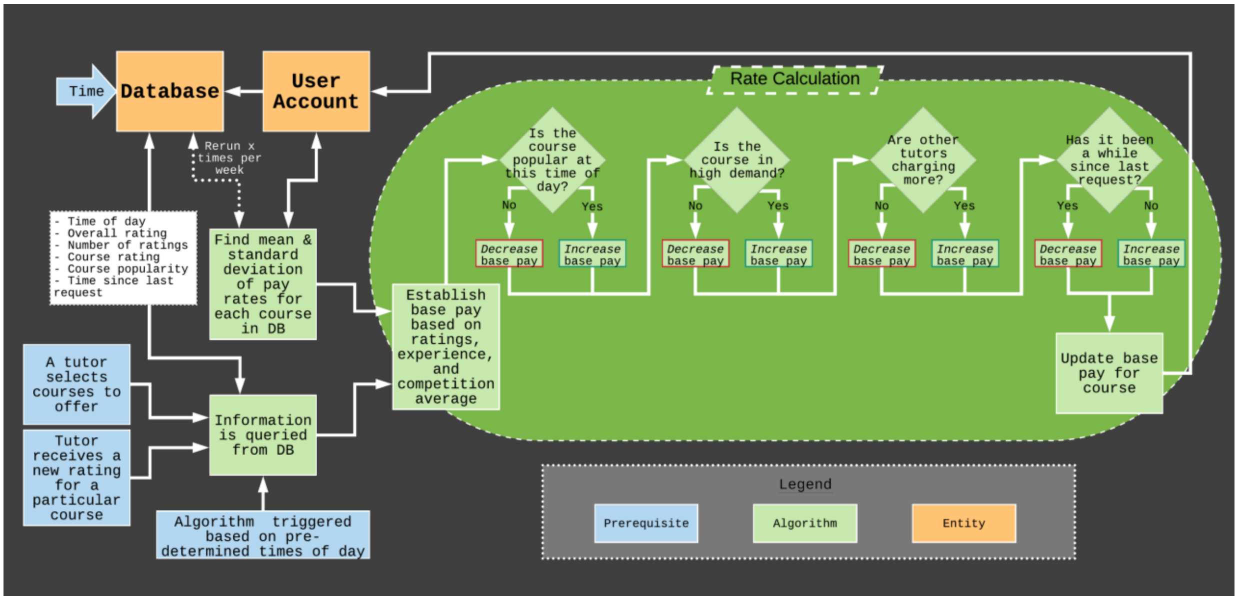
Time
Database
User Account
Rerun x times per week
- Time of day
- Overall rating
- Number of ratings
- Course rating
- Course popularity
- Time since last request
Find mean & standard deviation of pay rates for each course in DB
A tutor selects courses to offer
Tutor receives a new rating for a particular course
Information is queried from DB
Algorithm triggered based on pre-determined times of day
Establish base pay based on ratings, experience, and competition average
Rate Calculation
Is the course popular at this time of day?
No
Yes
Decrease base pay
Increase base pay
Is the course in high demand?
No
Yes
Decrease base pay
Increase base pay
Are other tutors charging more?
No
Yes
Decrease base pay
Increase base pay
Has it been a while since last request?
Yes
No
Decrease base pay
Increase base pay
Update base pay for course
Legend
Prerequisite
Algorithm
Entity

Figure 12 - Competitive Pay-Rate Algorithm Flow

Figure 12 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.

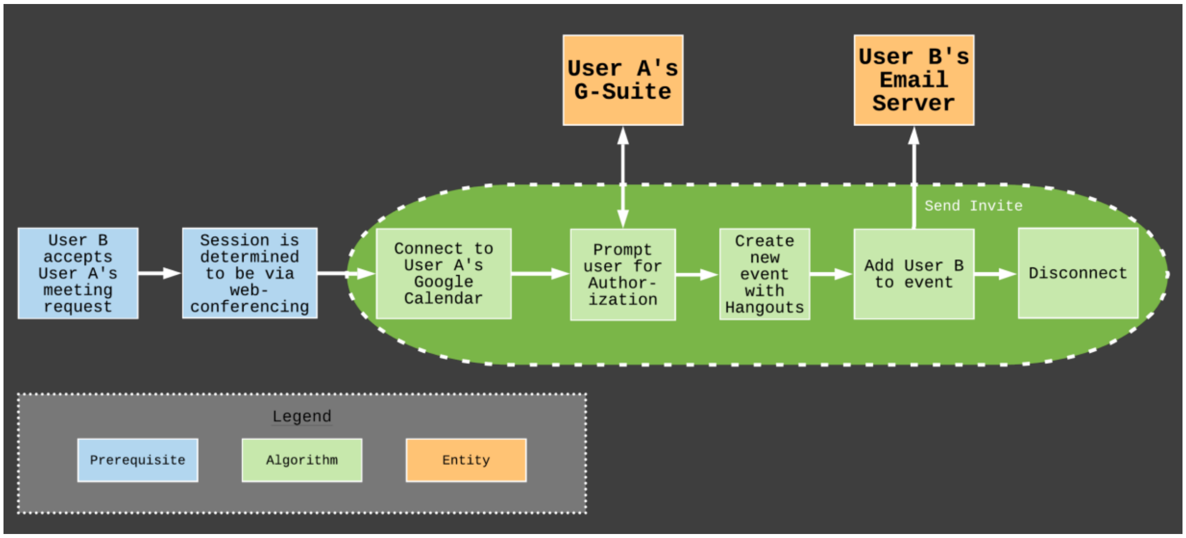
User A's G-Suite
User B's Email Server
Send Invite
User B accepts User A's meeting request
Session is determined to be via web-conferencing
Connect to User A's Google Calendar
Prompt user for Author-ization
Create new event with Hangouts
Add User B to event
Disconnect
Legend
Prerequisite
Algorithm
Entity

Figure 13 - 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 13 illustrates the flowchart of the algorithm.

[This space intentionally left blank]

The payment logic algorithm works in tandem with the pay-rate calculator algorithm to handle the distribution of funds from the tutee to the tutor, as depicted in Figure 14.

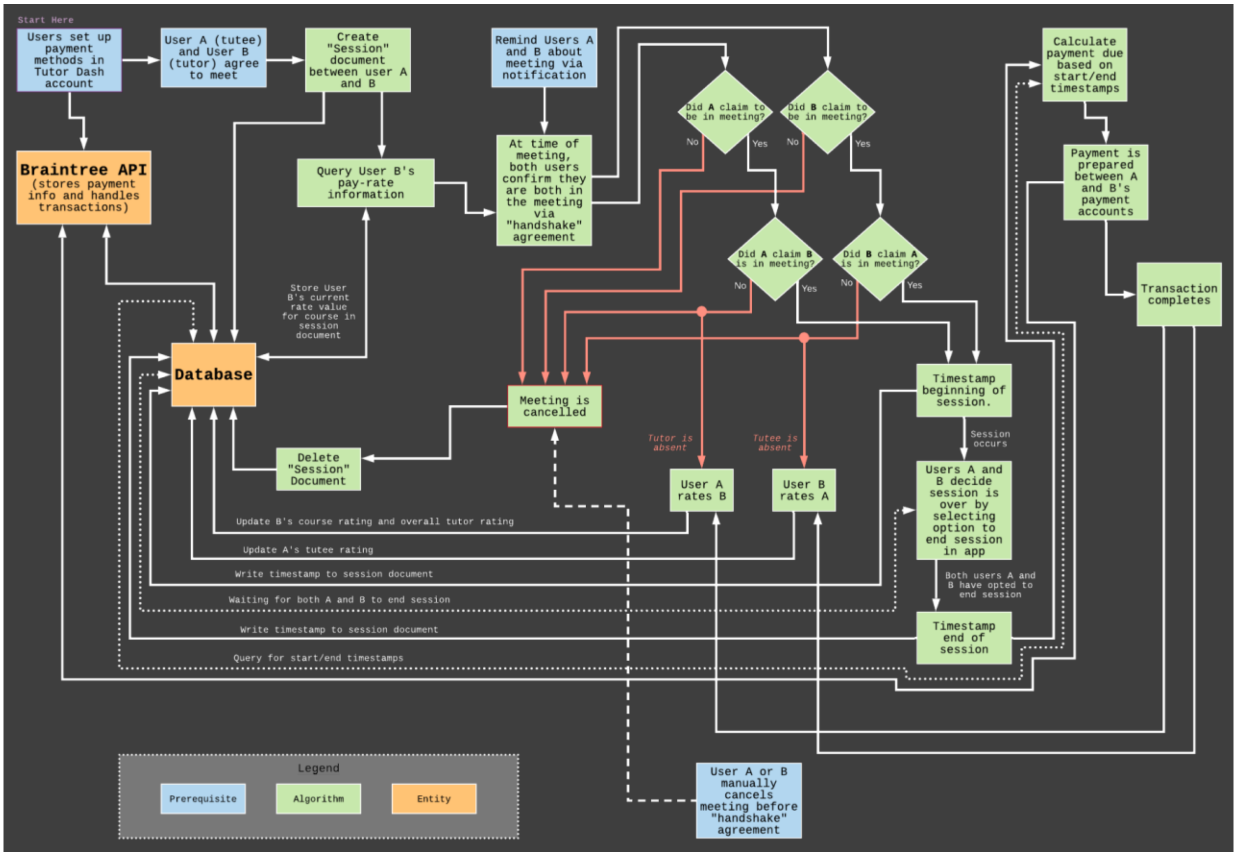
Start Here
Users set up payment methods in Tutor Dash account
User A (tutee) and User B (tutor) agree to meet
Create "Session" document between user A and B
Remind Users A and B about meeting via notification
Braintree API (stores payment info and handles transactions)
Query User B's pay-rate information
At time of meeting, both users confirm they are both in the meeting via "handshake" agreement
Did A claim to be in meeting?
Did B claim to be in meeting?
No
Yes
No
Yes
Calculate payment due based on start/end timestamps
Payment is prepared between A and B's payment accounts
Did A claim B is in meeting?
Did B claim A is in meeting?
No
Yes
No
Yes
Transaction completes
Store User B's current rate value for course in session document
Database
Meeting is cancelled
Timestamp beginning of session.
Session occurs
Tutor is absent
Tutee is absent
Delete "Session" Document
User A rates B
User B rates A
Users A and B decide session is over by selecting option to end session in app
Update B's course rating and overall tutor rating
Update A's tutee rating
Write timestamp to session document
Both users A and B have opted to end session
Waiting for both A and B to end session
Timestamp end of session
Write timestamp to session document
Query for start/end timestamps
Legend
Prerequisite
Algorithm
Entity
User A or B manually cancels meeting before "handshake" agreement

Figure 14 - 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.

2.3 External Interfaces

Tutor Dash does not require any external interfaces outside of an Android device or emulator to run the application. Development is handled mostly in Android Studio and external APIs (including Firebase Firestore, Firebase Authentication, Google Maps, Google Calendar, and Braintree) are utilized where they are necessary.