Lab 1: Tutor Dash Description

Brandon Campbell

CS 411

Professor T. J. Kennedy

26 September 2019

Version 1

Table of Contents

List of Figures

List of Tables

1. Introduction

University campuses provide a unique market with unique challenges. Students are often in need of tutoring services while, at the same time, in a position to offer tutoring services of their own. On a large campus, it can be difficult for students to communicate their need for services as well as promote tutoring services they may be offering.

Tutor Dash is a mobile application providing a digital marketplace for students to promote their tutoring services to other students. It provides a structure to ensure transparency on services offered through a built-in rating and review system and enforces a fair cost of services through the use of pricing algorithms.

Students seeking a tutor in a specific course often run into many obstacles. First, many university tutoring centers only offer tutoring services in courses that have high fail rates. If a student is seeking assistance in a course that is not offered, they rely on their peers who have taken that course. The obstacle is in finding that peer.

This leads to the second point. There is not a centralized social platform catered to connecting students for tutoring services. Figure 1 further outlines some of the obstacles that students run into while seeking a tutor and Figure 2 outlines the obstacles tutors have while seeking tutees.

[This space intentionally left blank]
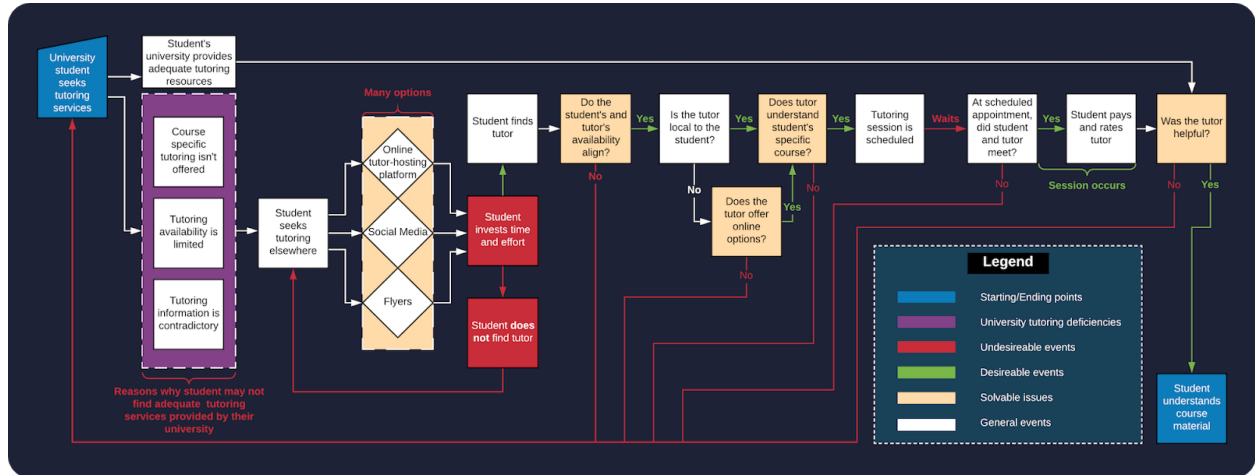
Figure 1. Students Seeking Academic Assistance

This is the current process flow for tutees looking for tutors. The diagram illustrates

many of the decision obstacles that make it difficult for students to find compatible tutors.
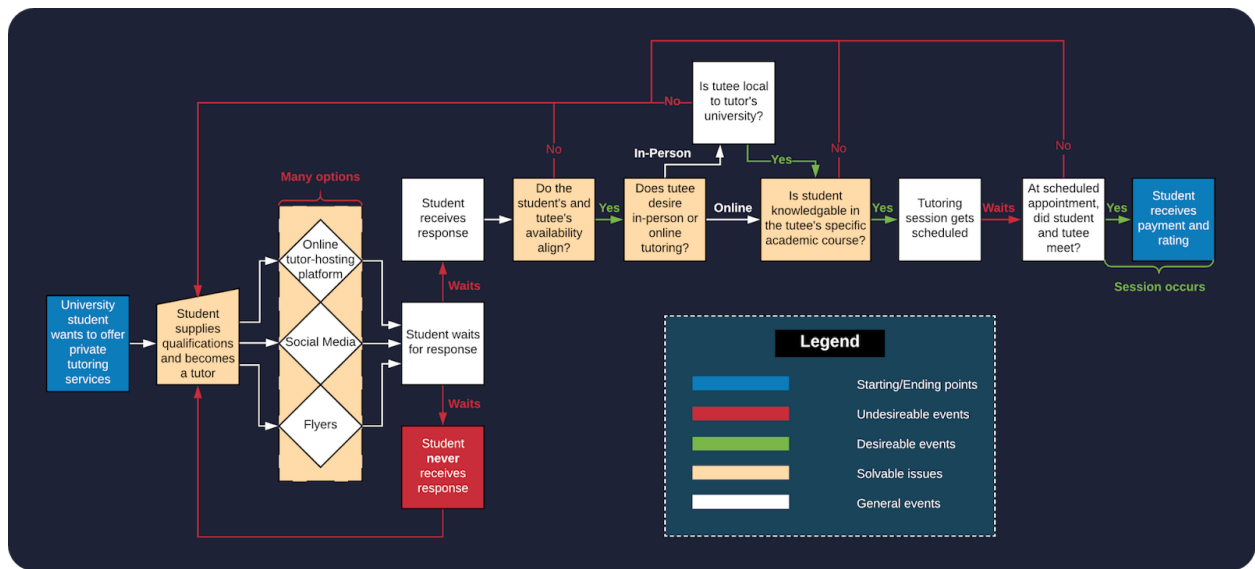
Figure 2. Students Seeking Tutoring Advertisement

This is the current process flow for tutors looking for tutees. Tutors looking for tutees

face similar obstacles to tutees looking for tutors as shown in Figure 1.

These figures summarize how the current process of finding tutoring services provides many challenges to students. Providing a central platform with the tools that connect tutors and tutees and make the transaction process simpler will mitigate these challenges. Providing easier access to academic assistance will decrease DFWI rates benefiting both the students and the universities.

2. Tutor Dash Product Description

Tutor Dash will streamline the process of connecting qualified tutors to tutees. It does this by eliminating many of the obstacles students face (Figure 3). Its benefit over the existing process is that it is designed specifically for tutoring services. While there are many ways to advertise services, they are not tutoring specific. When a student is in need of tutoring, they will be able to open the Tutor Dash application and have access to all of the tutors at their school. They can search this list of tutors by the courses they offer, their cost for services, and the tutor's rating.

[This space intentionally left blank]
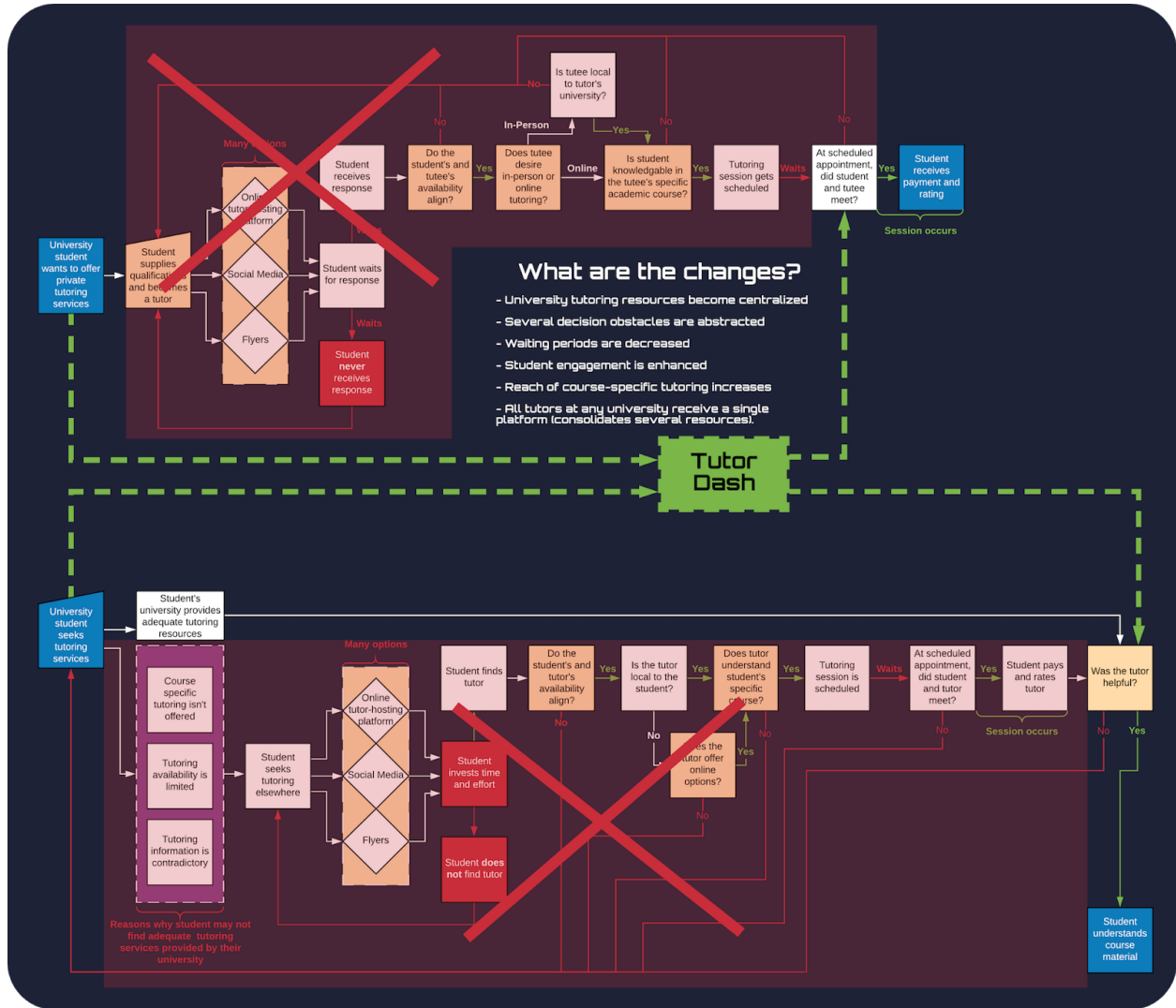
Figure 3. How Tutor Dash Affects the Current Processes

Tutor Dash merges the two current process flows illustrated in Figures 1 and 2. By providing a niche tool that connects students at the source, many of the current decision obstacles are bypassed.

Once the student selects the tutor that meets their needs, they can schedule a session and message the tutor directly. The tutoring session can be an in-person session or a web-based video session. At the conclusion of the session, the student will be charged for the services within the

application and will be prompted to rate and review the tutor. Likewise, the tutor will receive

payment for the services and will be prompted to rate and review the tutee.

      To become a tutor, a student must upload their transcript in the application. A parsing

algorithm will then pull the courses and grades from the transcript to identify which courses they

are qualified to tutor in. A course where a student received a B or higher qualifies. Transcripts

are deleted after parsing. The whole process can be seen in Figure 4 below.
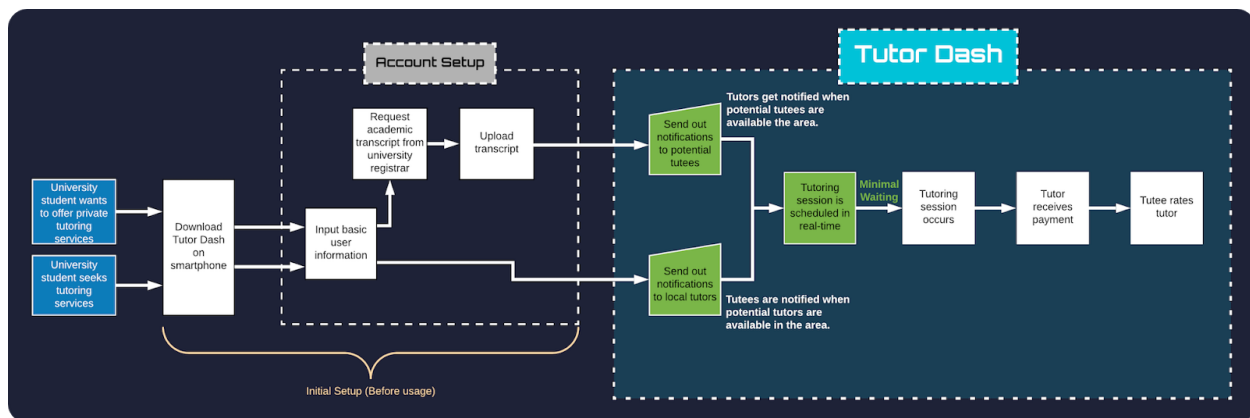
Figure 4. Tutor Dash Process Flow

      Tutors and tutees must download the app, and then sign up for an account. Once

membership has been established, users will then be notified about other potential compatible

users based on their login status (i.e., logged in as tutor or logged in as tutee).

2.1 Objectives and Goals of Tutor Dash

      The objective of Tutor Dash is to make it easier for college students to find qualified

tutors amongst their peers at an affordable price. It will fill the gaps in the market which

currently only consists of university resources, private tutors, and online tutoring businesses.

This will ultimately lead to more successful students and establish Tutor Dash as the go-to

resource for finding tutoring services.

The goal is to make the application simple, intuitive, and reliable so that students will be more likely to use the application and recommend it to their peers. Tutor Dash's modern approach to addressing the tutoring market will help break down the stigma involved in asking for academic assistance. As Tutor Dash grows in popularity, it will be scaled to work at any university.

2.2 Features and Capabilities

The users of the application will be limited to college students. This is to guarantee that the community of tutors and tutees are limited to a peer group. When creating an account users must provide a ".edu" email address. A verification email will be sent in order to activate the account. Users can only use the application by logging in with a verified account.

Users will be able to choose if they would like to be a tutor. All users are tutees by default. If a user elects to be a tutor, they will be prompted with a request to upload a transcript. After uploading their transcript, a parsing algorithm will determine which courses they are eligible to tutor. The transcript parsing will happen in real time allowing users to start using the application immediately.

In order to ensure that tutees are receiving the best quality services, the application will implement a rating and review system. Tutees can rate and write a review of tutors after completing a session. When searching for tutors, tutees can view ratings and reviews in order to select the most qualified tutor. This will help foster a competitive market.

Tutors will have a feature to set their availability. If a tutor indicates that they are currently available, then tutees will know who can provide immediate services. If a tutee does not find any available tutors, they will have the option of sending out a request to all tutors who

are eligible to tutor their desired course. Those tutors may respond to the request and schedule a session with the tutee. Tutees may also send a request to a specific tutor, which the tutor may accept or deny.

The selection of courses that tutors will be capable of tutoring is only limited to the list of course offerings at their respective school and the grade they received after taking that course. This is a main feature of the application and is what sets it apart from currently existing tutoring services. Tutor Dash will also allow students to schedule tutoring services 24/7. This is a feature that most schools will never be able to offer.

Another important feature is the in-app chat capabilities. The goal of Tutor Dash is to keep all steps of the tutoring process within the application. There will be a built in messaging feature that will allow for real time communication between users. This is necessary for scheduling tutoring services.

Tutoring sessions can be scheduled as in-person meetings or a web-based meeting. When scheduling an in-person meeting, both parties will simply agree on a time and place. If a web-based meeting is desired, then both users will be redirected to a Google Hangouts meeting. This is one of the few features where the application will defer to another application.

Payments for tutoring services will be handled by a third party library from Braintree. Braintree provides the ability to integrate payment processing within a mobile application. Payment is provided by the tutee where Tutor Dash will take a small percentage and the remaining payment is passed on to the tutor. A tutor's hourly rate will be determined by an algorithm. This algorithm will be discussed further in section 4.2.3.

2.3 Major Functional Components

   The major functional components of Tutor Dash consist of hardware, software, and

entities. The entities mainly consist of the end users and their email addresses. These entities will

utilize the software which makes up Tutor Dash via a smart device (hardware). Figure 5 shows

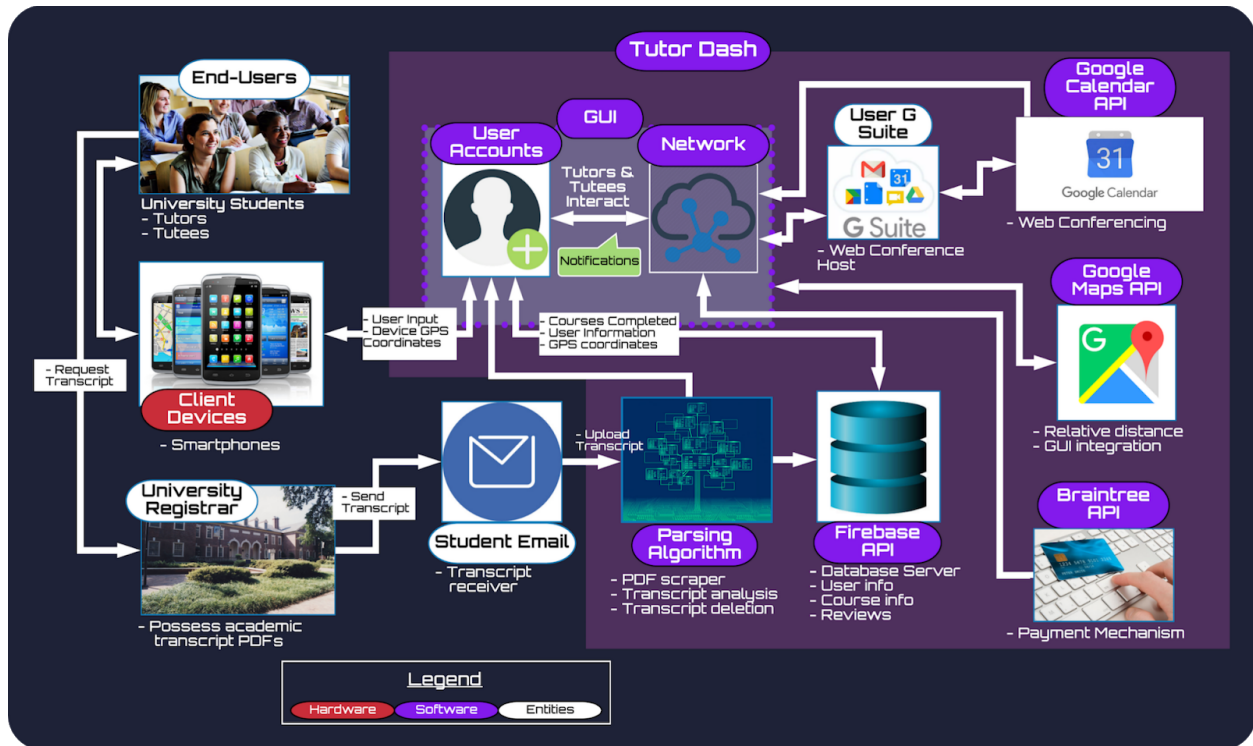what components are involved and their relationship to each other.

Figure 5. Major Functional Component Diagram

   Tutor Dash has an extensive back-end. To do this, it will leverage several

well-documented APIs suited for Android such as Firebase, Braintree, Google Maps, and Google

Calendar.

2.3.1 Software. The software components consist of a GUI, user accounts, parsing

algorithms, and third party APIs. These components together provide the functionality of Tutor

Dash.

The user's main interaction with the application will happen through a GUI. This consists

of the layouts that are displayed on the mobile device. For Android devices, the GUI will be

developed in XML as part of the Android SDK.

In order to use the application, users must create a user account. The user account

contains all of the information about that user and may limit what features the user is capable of

using. The user's information will be stored in the database and all accounts will be authenticated

through Firebase Authentication. Firebase is a collection of development tools for backend

development. The database will also be handled through Firebase. The Firebase database, Cloud

Firestore, is a noSQL, JSON-like, document based database. It provides real time database

interaction.

[This space intentionally left blank]

2.3.2 Hardware. The hardware required for Tutor Dash is a mobile device. For the initial development, the mobile device will be an Android smartphone.

2.3.3 Entities. The entity components consists of the end users, email addresses, and GSuite. The end user will consist of tutors and tutees. Their roles have been discussed extensively in earlier sections. The user's email address will be necessary for account creation. Only users with a college email address will be able to create an account. GSuite is necessary for web-based tutoring sessions. If users would like to schedule web-based sessions, then they must have a valid Google account.

3. Identification of Case Study

The current customers for Tutor Dash are students who attend Old Dominion University. These are students who are either seeking tutoring services or seeking to offer tutoring services. Those looking to offer tutoring services are likely students who are already tutoring and would like a way to expand their customer base, or they are students who are just looking to earn some extra cash from time to time without commitments.

The school would be considered an indirect customer. They can stand to gain from Tutor Dash being used widely on their campus because it can help alleviate the burden on school provided tutoring services and help improve the performance of the student body. A school's DFWI rate is a main indicator on the quality of the education provided by the school. In the future, Tutor Dash seeks to expand to other educational institutions.

[This space intentionally left blank]

4. Tutor Dash Prototype Description

The initial development of Tutor Dash will be a prototype version. The prototype will be

limited in features and scalability when compared to the real world product (RWP). Table 1

below lists the differences in functionality between the prototype and the RWP.

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 real world product is not too much different than the development prototype as many of the proposed features will be fully functional in both. However, the prototype will need to have some data mocked up to simulate various user accounts.

The prototype is meant to provide a proof of concept and demonstrate the application's usefulness. Development of the RWP will require a lot of time, money, and resources. The prototype will demonstrate basic user interaction, transcript parsing for tutoring capabilities, rate calculation algorithm, and the ability to schedule tutoring sessions either in person or web-based.

4.1 Prototype Architecture

The architecture of the prototype will consist of the same components discussed in section 2.3. The main difference in the prototype is that some of those component's features will be limited.

[This space intentionally left blank]

4.1.1 Prototype hardware. The hardware used in the prototype will be limited to an

Android smartphone running the Android operating system. The RWP will be expanded to

include an iOS version which can be used on iPhone devices.

4.1.2 Prototype software. The Firebase database for the prototype will be limited in

storage size and the amount of queries that can be made per week. Firebase provides scalable

payment options which will allow growth for the RWP.

4.2 Critical Design Components

The Tutor Dash application is comprised of three major components: the database,

UI/UX, and algorithms.

4.2.1 Database. As stated, the database is Cloud Firestore which is provided through

Firebase. Since Firebase is a Google product, it will easily integrate with the Android application

and has the capability of extending to iOS and web applications. The prototype database will be

limited to 1 GiB of storage, 10 GiB/month of bandwidth, 20K/day writes, 50K/day reads, and

20K/day deletes.

The database will contain all the data about the users and the schools as well as contain

the data for in-app messaging. Table 2 illustrates the database schema.

[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

Unlike a traditional relational SQL database, Firebase does not utilize Entity-Relationship

Diagrams. This figure shows each JSON document as it exists in the Tutor Dash prototype.

[This space intentionally left blank]

4.2.2 UI/UX. The UI/UX of the application is important in getting users to use the application and navigate it without problems. Figure 6 below demonstrates a wireframe modeling of the UI/UX. Android provides pre-built interface components which are organized in an XML file to create layouts. The prototype will be limited to simple design components while still providing an effective UX.
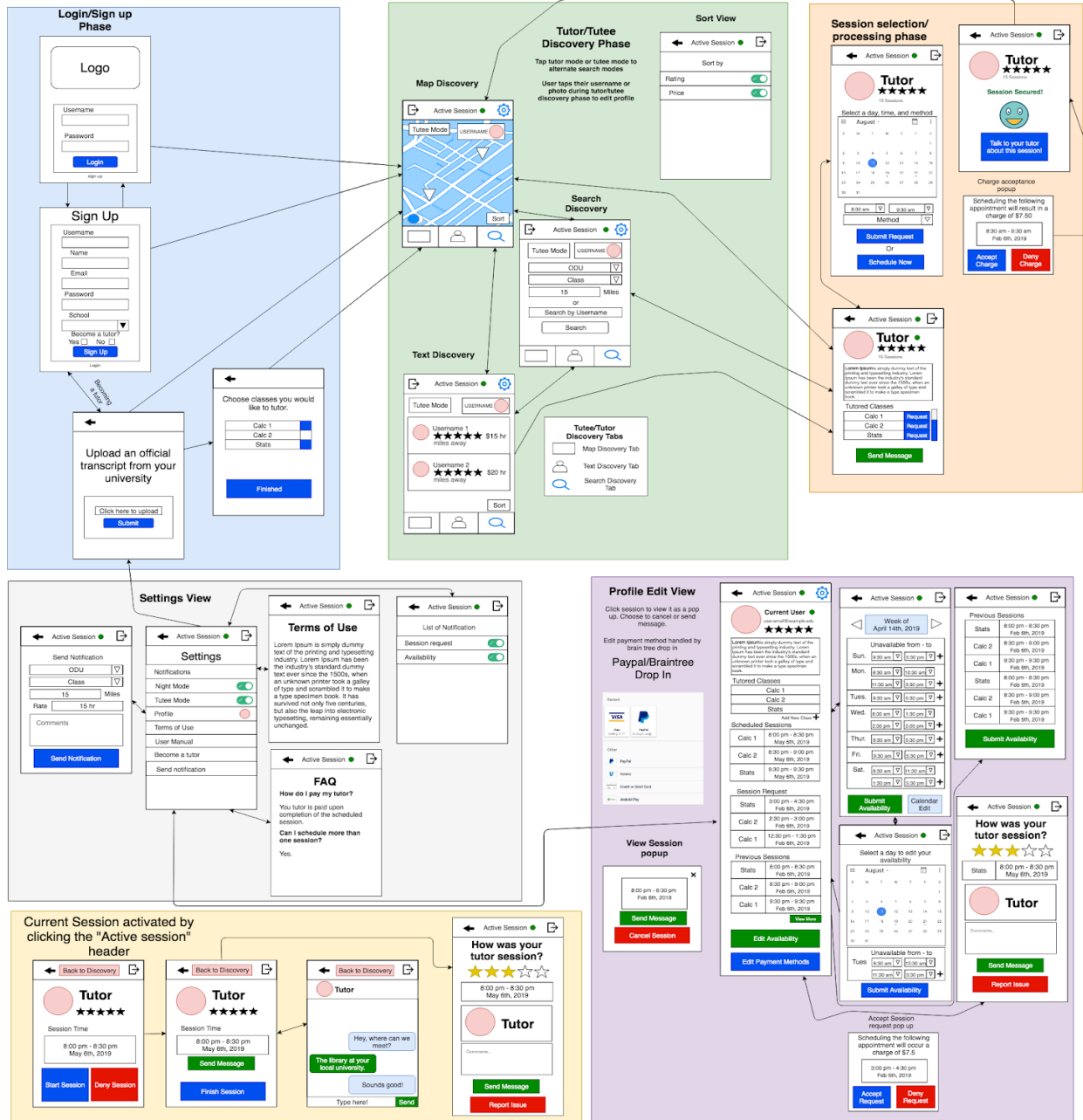
[This space intentionally left blank]

Figure 6. UI/UX Full Implementation

The UI/UX is composed of six phases and two views. Phases are areas of the UI/UX that

are visited in a specific order, while views are areas of the UI/UX that are not visited in any

specific order.

Figure 7 to Figure 11 gives a closer look at the different phases of the user interface. Each phase encapsulates a collection of layouts that are necessary for a specific activity. For example, Figure 8 shows the main landing page of the application. It's the first layout the user will see after signing into the application. From here, the user can navigate to the main features of the application.
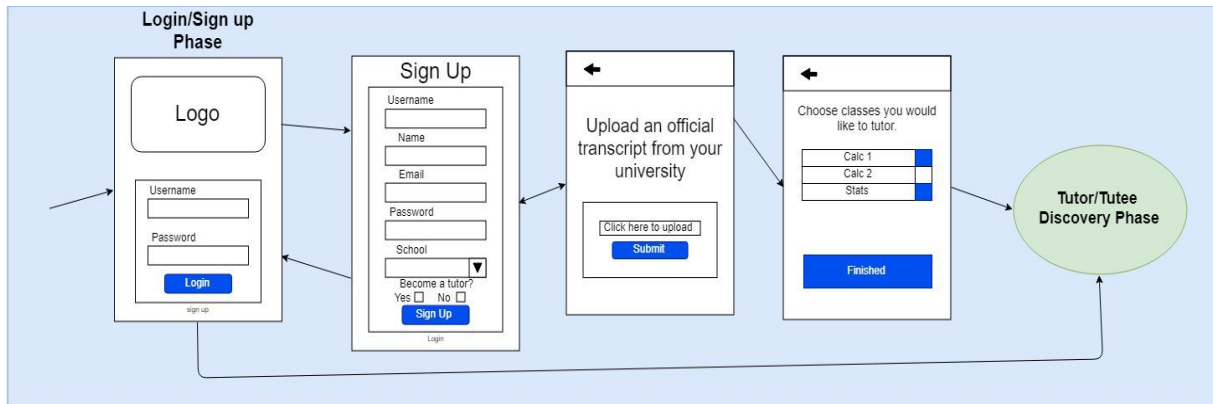

Figure 7. UI/UX Phase 1

This figure illustrates the login/sign up phase. During this phase, the user will either sign up for a new account or log in with an existing account. If they are signing up and wish to become a tutor, then they can provide their transcript during this phase if desired.

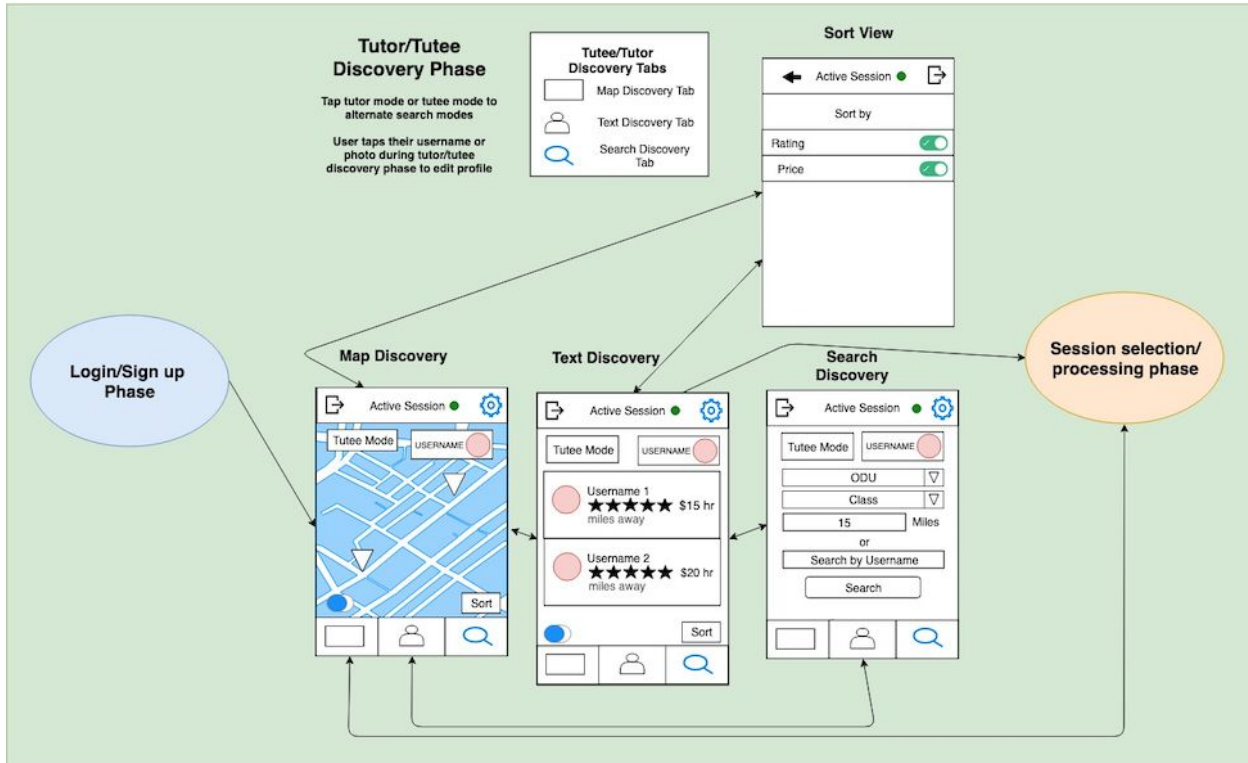[This space intentionally left blank]

Figure 8. UI/UX Phase 2

This figure illustrates the tutor/tutee discovery phase. During this phase, users will be able to interact with the database by searching for other users based on their own search parameters.

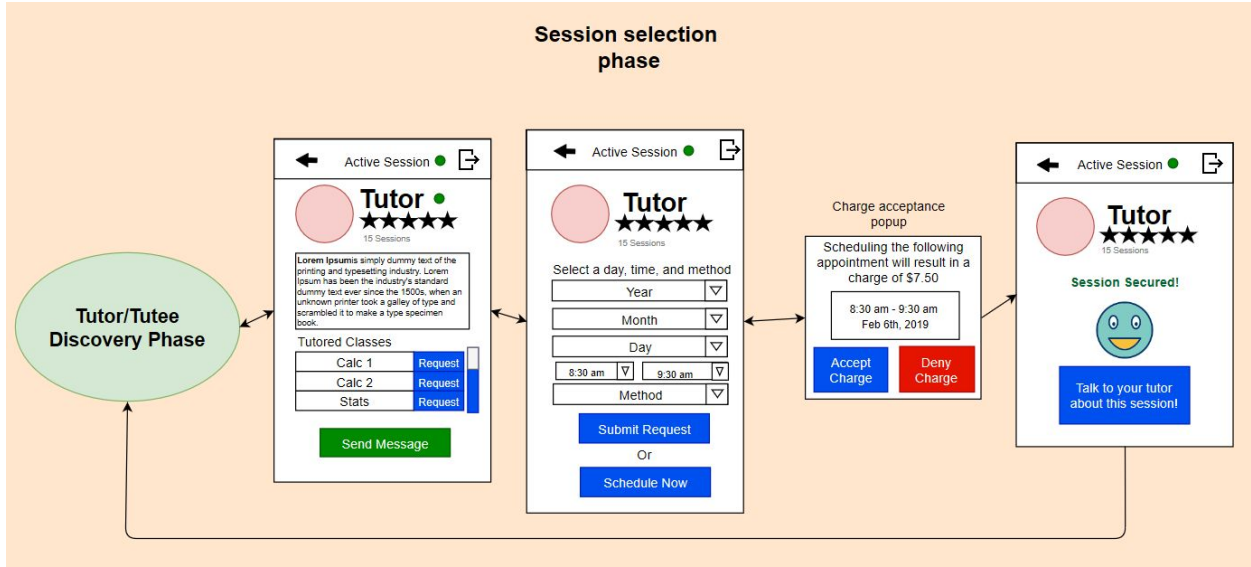[This space intentionally left blank]

Figure 9. UI/UX Phase 3

This figure illustrates the session selection phase. During this phase, the user will create a request for a session that fits theirs and the tutor's schedules. Once this request is sent, the tutor will be notified.
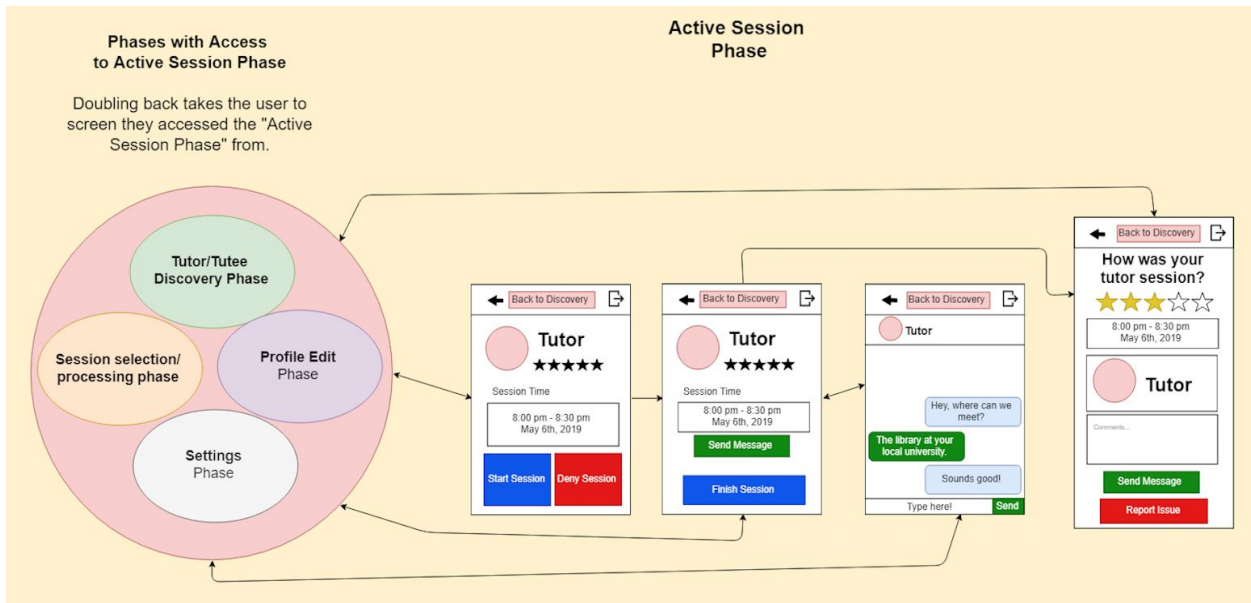

Figure 10. UI/UX Phase 4A

This figure illustrates the active session phase. When a session has started, both tutor and tutee will be prompted to acknowledge that both users are in the meeting. The session will then proceed with a timer, and after the session ends, the tutor will be paid. Both users will then be prompted to rate the corresponding user.
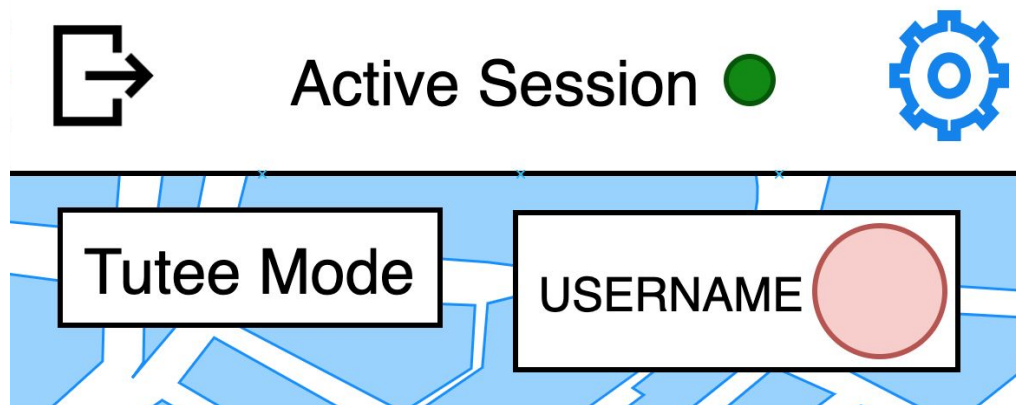

Figure 11. UI/UX Phase 4B

This figure illustrates an important feature of the active session phase which is the ability for users to toggle their availability manually, but they will become unavailable during a session. When compatible tutors become available, tutees are notified, and vice versa.

Figure 12 and 13 illustrate the two views within the application. A view, as opposed to a phase, is a layout that allows the user to customize their settings and edit their personal profile information.
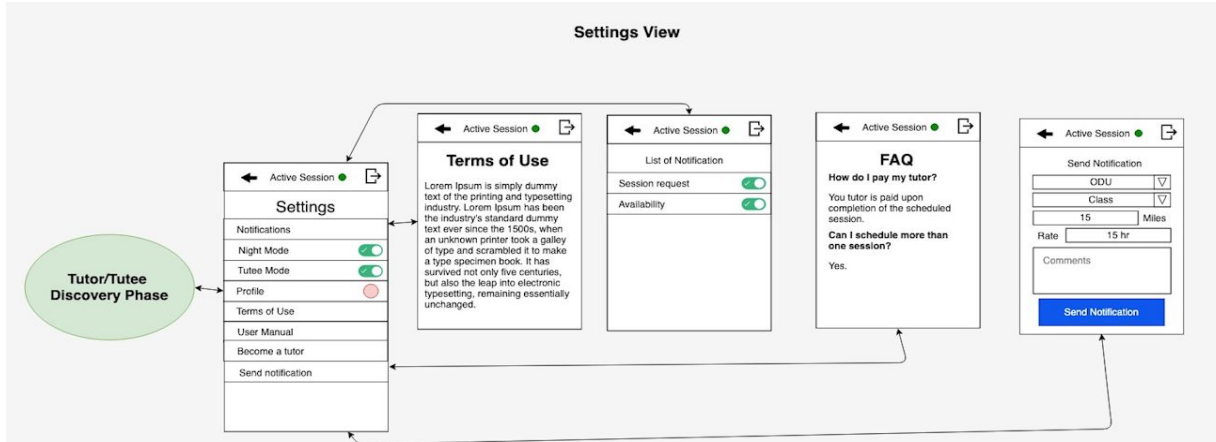
[This space intentionally left blank]

Figure 12. UI/UX View 1 - Settings

The settings view gives users the ability to modify notifications, toggle night mode, view their calendar, view their profile, review the terms of use agreement, view user manual, sign up to become a tutor, and send out local availability alerts.
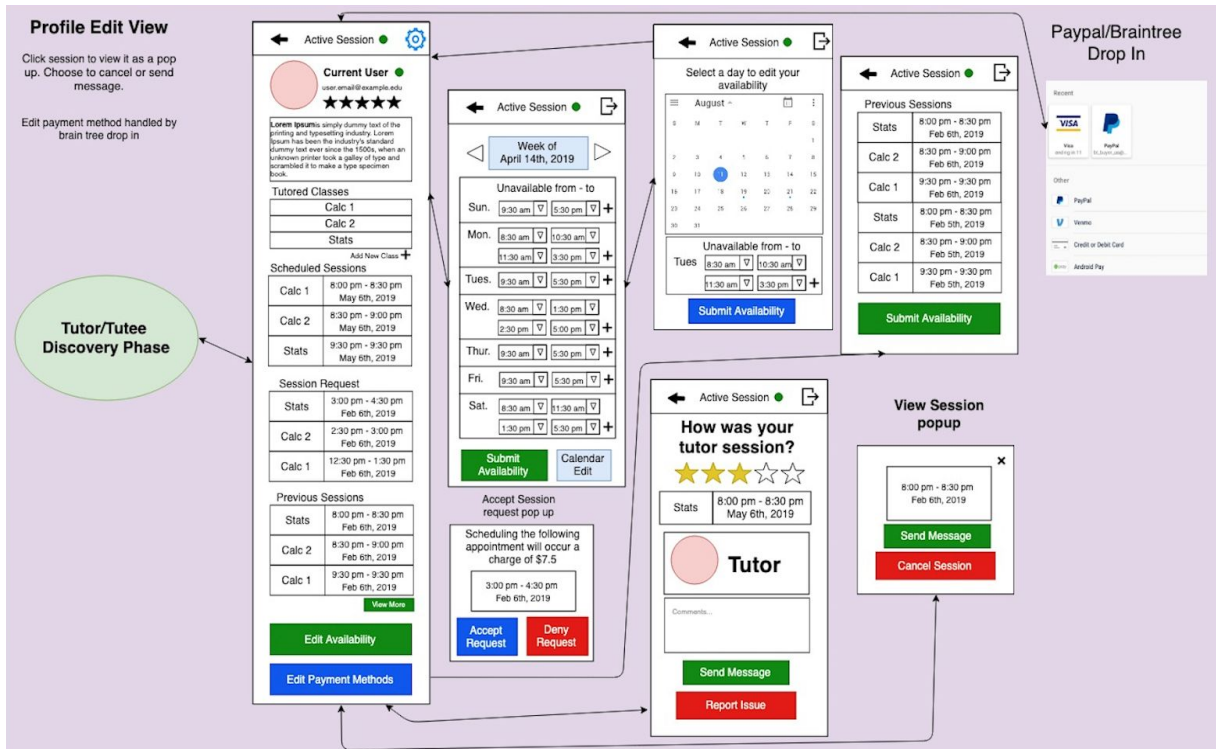

Figure 13. UI/UX View 2 - User Profile

The profile edit view allows the user to view and/or modify public and private information stored in their user profile. All information is viewable, but not all information on the profile is able to be modified (e.g., rating, previous sessions).

4.2.3 Algorithms. The following algorithms will provide the data manipulation necessary for the functionality within the application.
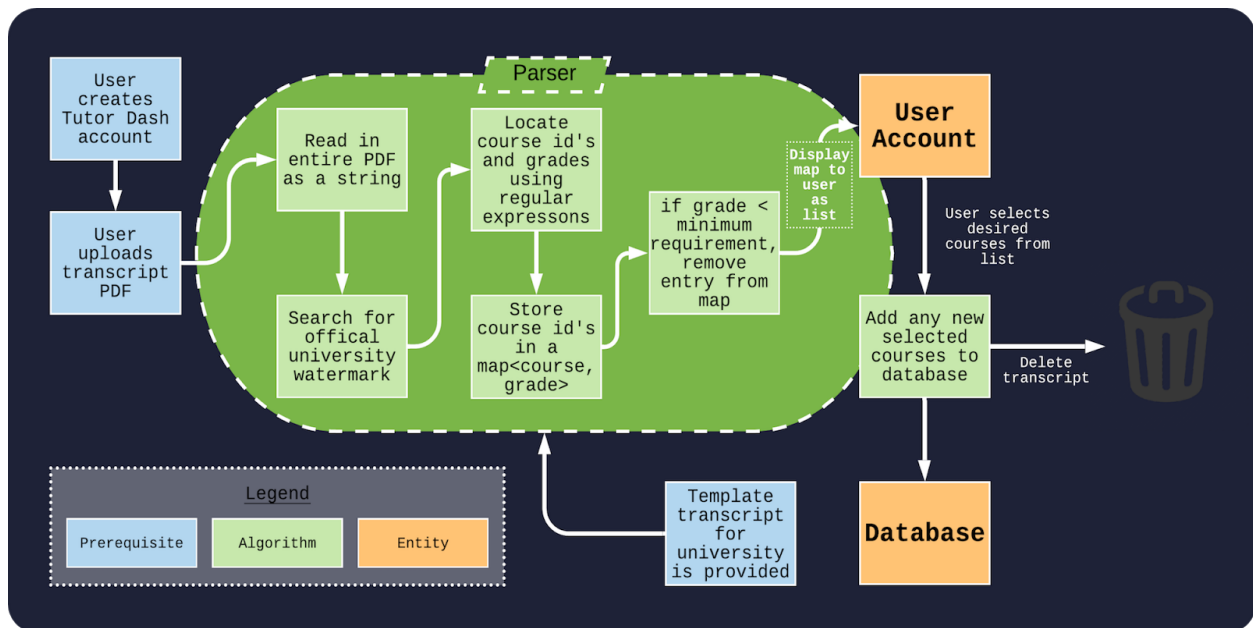

Figure 14. PDF Transcript Parser Algorithm Flow

This figure illustrates the PDF transcript parsing algorithm. Transcripts will be parsed as PDFs, and if they are valid transcripts, the user will be qualified to tutor selected courses based on their previous grades present in the text.

[This space intentionally left blank]
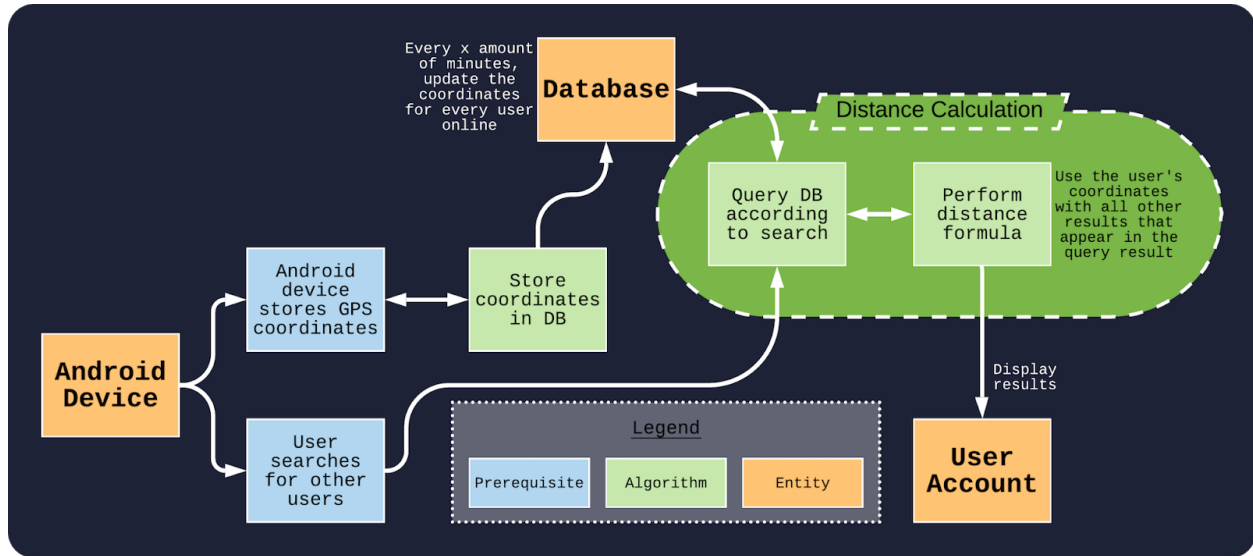
Figure 15. Relative Distance Calculator Algorithm Flow

This figure illustrates the relative distance calculation algorithm. Coordinates from

Android devices will be stored in the database and updated frequently. By using the distance

formula, an estimate for distance away other users are from the current user is calculated.
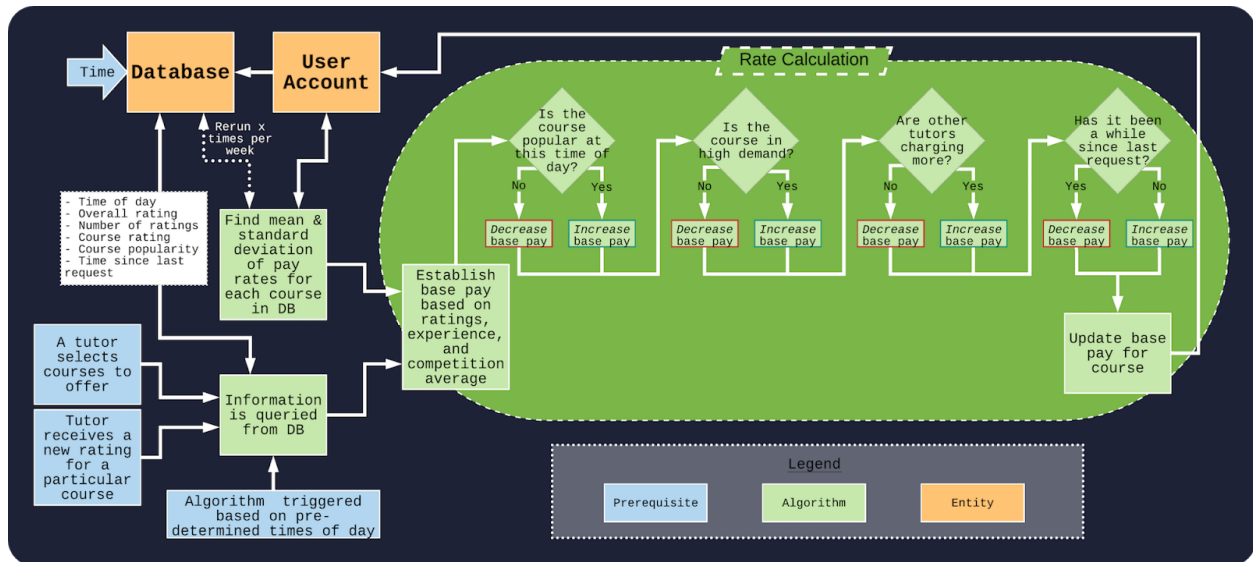
Figure 16. Competitive Pay-Rate Algorithm Flow

This figure illustrates the pay-rate calculation algorithm. Users will not set their pay, rather an algorithm will determine their pay based on course popularity, course demand, other tutors' pay-rates, and time since last request for course tutoring session.

Figure 17. Web-Conference Appointment Creator Algorithm Flow

This figure illustrates the web-conference event creator algorithm. If a session is determined to be via online, then an event with Google Hangouts conferencing will be created on the tutor's Google Calendar and sent to the tutee automatically.

[This space intentionally left blank]
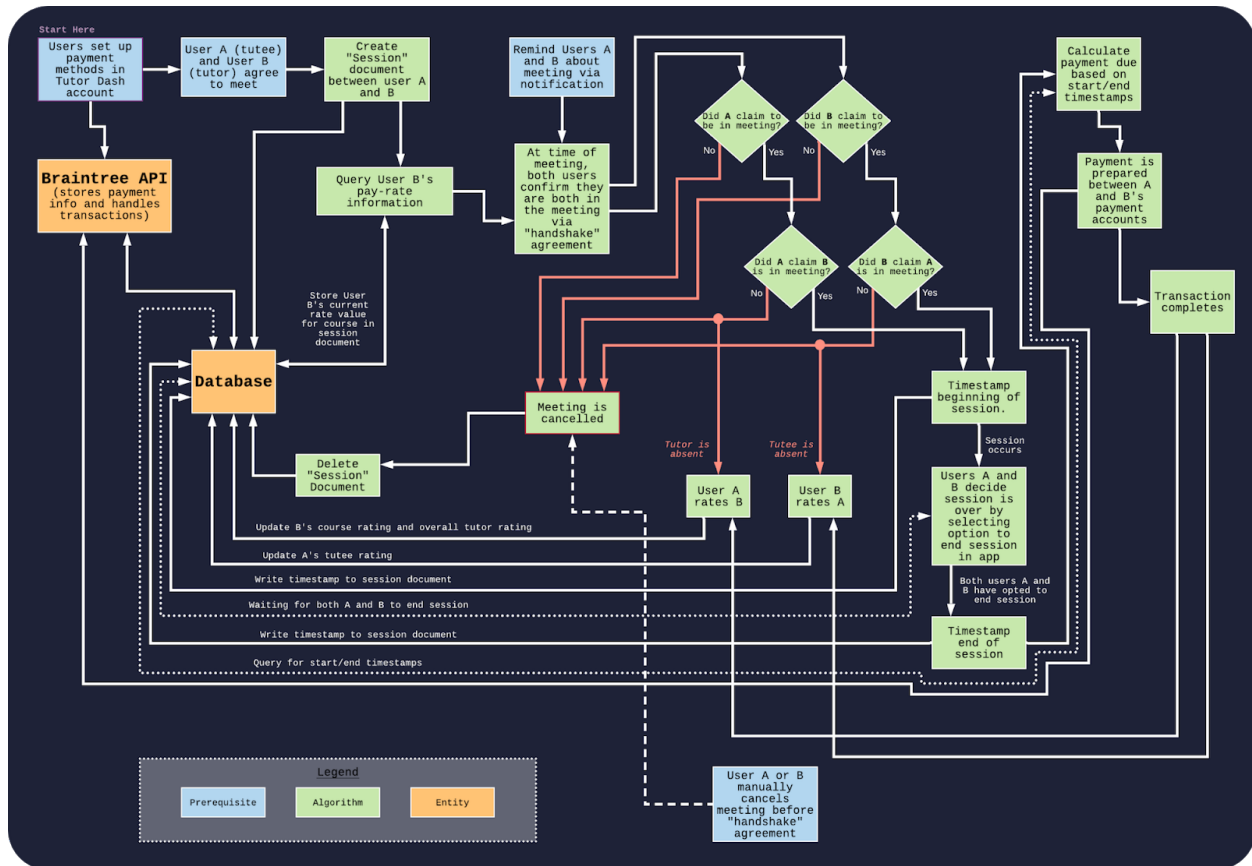
Figure 18. Payment Logic Algorithm Flow

This figure illustrates how payments will be handled in Tutor Dash. To improve security,

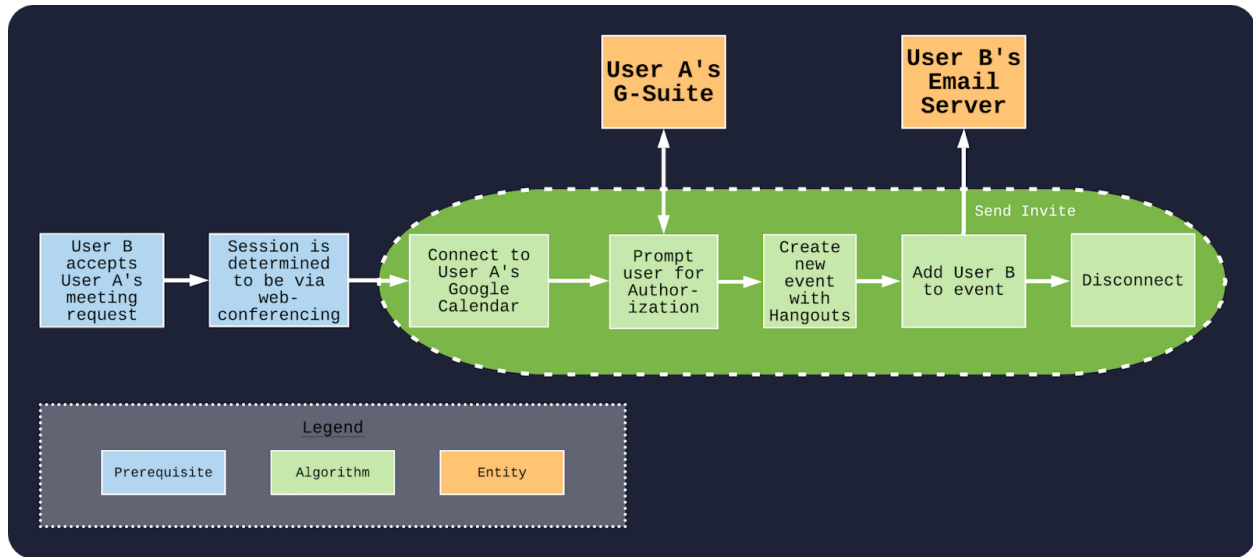payment accounts will be handled separately by a reliable and well-documented API called

Braintree.

4.3 Prototype Features and Capabilities

The features and capabilities of the prototype will only be slightly limited when

compared to the RWP. The main features, such as account creation, session scheduling, in-app

messaging, and web conferencing will be fully functional. Features such as user reporting,

blacklisting, and refunds/monetary incentives will either be limited or non-functional. This is

mostly due to financial and resource constraints.

The prototype payment rate algorithm will also not be as complex as expected in the RWP. The mean and standard deviation used in the calculation will be mocked up since the prototype will not have active users to establish them. Table 1 outlines the limitations of the prototype features more fully.

4.4 Risks and Mitigations

Figure 19 details the potential risks and mitigations that may occur with the application. Many of the risks are mitigated by implementing good software development strategies. Some risks, however, do not have good technological or design solution. Some of the higher risks, C3 and C4, are a good example of this. The application cannot be successful if it doesn't have an adequate amount of both tutors and tutees. The solution provided to mitigate this risk is to provide financial incentives for users to sign up and increase payment rates for high demand or under served courses. Figure 20 lists the potential risks and mitigations in more detail.

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

This figure illustrates all of the risks that the Tutor Dash RWP faces. Each of these risks is assigned an id that can be referenced 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
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

This figure describes the risks for the Tutor Dash RWP introduced in Figure 19. Some of these risks such as C3 and C4 will not be mitigated in the prototype since they are mainly concerned with quantity of users.

4.5 Prototype Objectives and Goals

The objective of the Tutor Dash prototype is to demonstrate that the application provides suitable means for establishing a controlled, easily accessible, and robust marketplace for tutoring services. This will be achieved by demonstrating the functionalities of the applications with a limited user base.

4.6 Prototype Development Challenges

Development of the prototype presents many challenges. The most pressing challenge is the timeframe that the application must be developed in. The timeframe is approximately four months which doesn't leave room for roadblocks or steep learning curves.

Another challenge is the team's experience with the tools and technology being used for development. The majority of the team has no Android or mobile development experience and has not worked with many of the third party APIs used in the application.

[This space intentionally left blank]

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. 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 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

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

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

   omprehensive-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=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

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=d93df6c4-37

    29-4b62-8d58-95e25c309878%40sessionmgr102&bdata=JnNpdGU9ZWhvc3QtbGl2ZS

    ZzY29wZT1zaXRl#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/s41239-018-010

    0-7

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://bmcm

    ededuc.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&needAc

    cess=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