Lab 1: Tutor Dash Description

John Hessefort

CS 411W

Thomas Kennedy

26 September 2019

Version 1

Table of Contents

1. Introduction

Tutor Dash will be implemented as a smartphone application for Android devices that is designed to centralize the tutoring services available on college campuses and allow student users to search for and engage with tutoring services provided by other students in attendance at the same university. Students who are in need of course assistance will be able to request help and schedule tutoring sessions with fellow students who through the application have decided to assume the role of a tutor for their peers.

The availability of tutors and tutoring services on most college campuses is lacking in comparison to the desire and need for them to aid in student success. Reasons for this include the facts that private tutoring is a difficult market to get into without proper connections, formal tutoring may not be able to suit the schedule of the student who wishes to help, and formal tutoring offered through an organization often has too many requirements for a student to feasibly fulfill when applying for the position. Allowing students to register as tutors through Tutor Dash will increase accessibility to the profession and thus encourage them to engage with peers in their campus community through the offering of help and guidance in a variety of courses.

It has been proven at ODU that the introduction of student peer tutoring services is highly beneficial. One such place in which this is seen is in courses pertaining to math and computer science, which many students will tend to struggle with even if only for certain topics or concepts. Moreover, for many courses and particularly with regard to MATH103M (College Algebra), the introduction of tutoring and supplemental instruction resources into the classroom setting has markedly improved the DFWI rates for the course and led to an increase in major retention among students. Through the improvement to DFWI rates many students have been

spared the expenses of having to retake a failed course and rely upon grade forgiveness policies. Many students have stated that they enjoy peer tutoring because they feel more relatable to and comfortable with the tutors than they do with their professors and graduate teaching assistants of the course. There is a formality expected with those relationships that is not present in peer tutoring, and it can at times pressure a student into not seeking help even when they genuinely feel that they need it for one or more of their courses.

Currently at ODU there is a wide variety of student benefit and tutoring resources available on campus, but because they are spread out across myriad departments information about the tutoring services may be rarely updated and can be difficult for students to locate online in a university's web domain, and students are less likely to make use of them as a result. Tutor Dash will connect students and consolidate these resources through the provision of a centralized platform for students to communicate with each other and give or request tutoring services whenever necessary or desired.

As was previously mentioned, many university students are qualified to aid their peers with course content but will often choose not to because of inconveniences with scheduling work hours around time necessary for school and the reception of payment for the work that they do helping peers. Tutor Dash will rectify this through the provision of a platform where students can offer or request tutoring help at any time of day, and tutors will receive payment for their services immediately upon completion of a tutoring session.

The services offered by the current university tutoring market do not suffice to fulfill the desires of students. They currently are very limited in scope; for example, ODU currently offers tutoring for only 132 courses when there are over 2400 courses available for registration. This means that ODU's existing tutoring services only cover approximately 5% of all ODU courses.

Over half of these available courses are under the 300 level. Additionally, the tutoring centers on

campuses are open for a limited range of hours every day and are not always available to

students when they may need them. There is great inconvenience to students that can be seen in

the fact that many tutoring resources are time-restricted to only several days per week or

available exclusively by appointment. While more of the tutoring centers are open during the

early afternoon, very few remain open until later in the afternoon and early evening, which

studies have shown is a period of time that many students prefer to study during. Scheduling

information for tutors and tutoring center open hours is not always up to date and reliable, so

oftentimes students may arrive at one to request help and soon discover that there is no tutor

currently present or available to assist them. On the following page can be seen Figure 1 which

features a process flow diagram depicting the current procedure for locating a tutor, requesting

help with coursework and understanding class concepts, and receiving said help from the tutor.

Furthermore, the second diagram in Figure 2 covers the process that a university student must go

through in order to attempt to begin private tutoring. 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. The diagram illustrates many of the decision obstacles that make it difficult

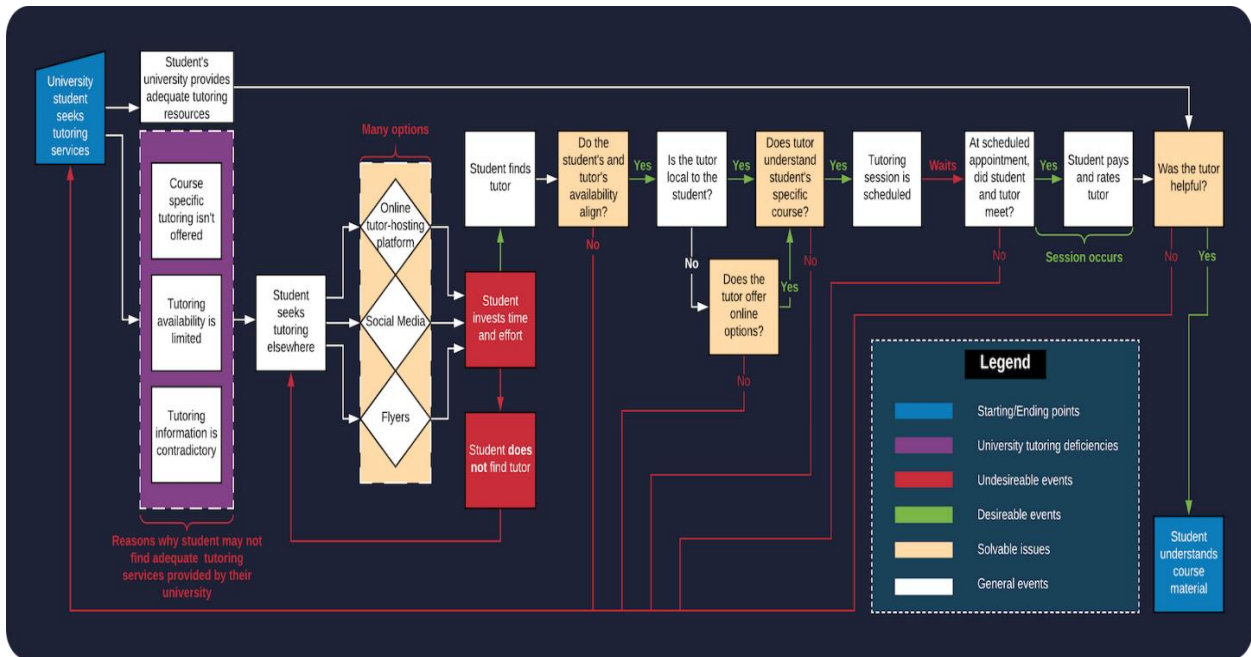for students to find compatible tutors.

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.

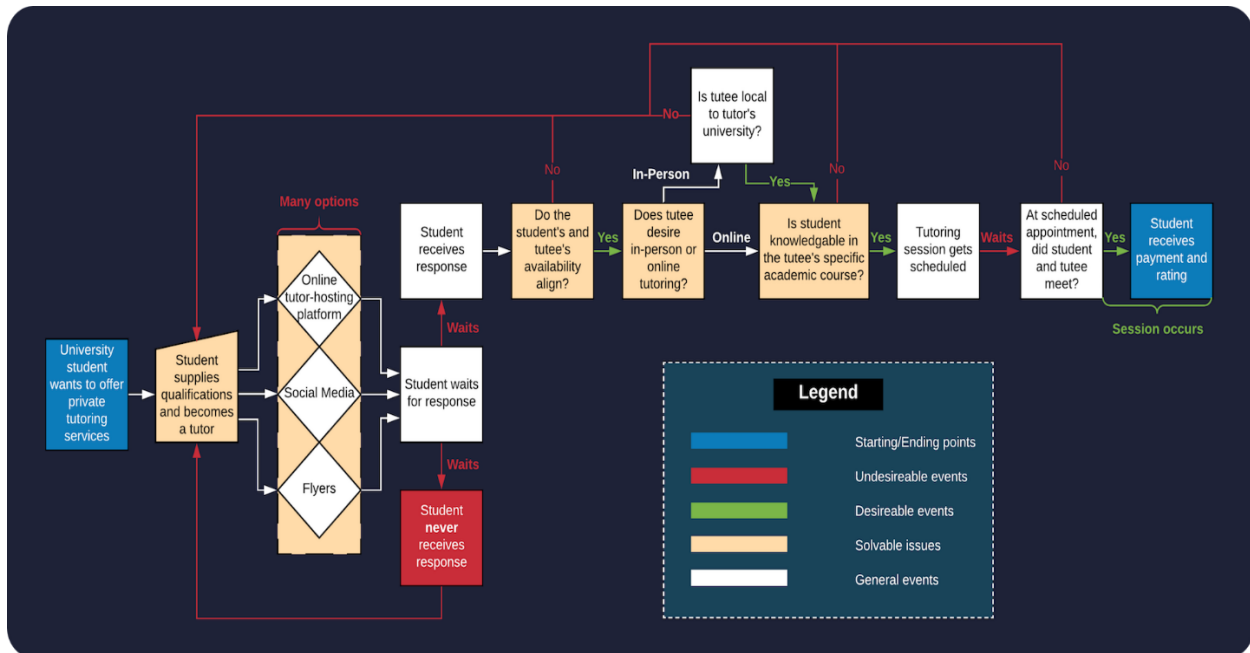[This space is intentionally left blank.]

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.

Tutor Dash will allow qualified students to tutor any class at their university, and they as

tutors can make themselves available at any time of day that is convenient for them. Tutor Dash

consolidates tutoring services onto a single platform, regardless of whether or not a student

works for the university itself at a tutoring center or offers private tutoring services. Tutoring will

be available both in-person and online for the entirety of each day, and any information

regarding a university's tutoring resources can be maintained more conveniently and in real-time

through the implementation of a check-in and check-out system that university tutors can use to

display accurate information for tutees to peruse when determining if there is a tutor available for

the course that they wish to be tutored in. Every tutor will be required to provide transcript

verification indicating that they are qualified to tutor in any particular course that they wish to

assist others with. Only university students will be able to use Tutor Dash, and this will be

ensured through the implementation of a system that requires users to have a university-affiliated

email address when creating an account. For the convenience of tutees and tutors Tutor Dash will support the capability for tutors to provide their location and relative distance from tutees. Lastly, payments for tutoring services will take place directly within the application and will not require the active use of any third-party applications.

2. Tutor Dash Product Description

Tutor Dash seeks to increase for students the convenience in being both tutees and tutors through the centralization of existing campus tutoring resources, ease of student qualification and application to becoming a tutor, and the provision of more tutoring services. Students end-users will make use of the application at any time through the offering to student tutees of tutoring services for the courses at their university, the requesting of such services from student tutors, or both. This adds convenience to the process of tutoring for tutees by allowing them the possibility of receiving help at any time of day; moreover, it adds convenience for tutors by serving as a platform through which they can advertise and offer their services, which before was only possible primarily through the posting of flyers around campus and the printing out of business cards. 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.
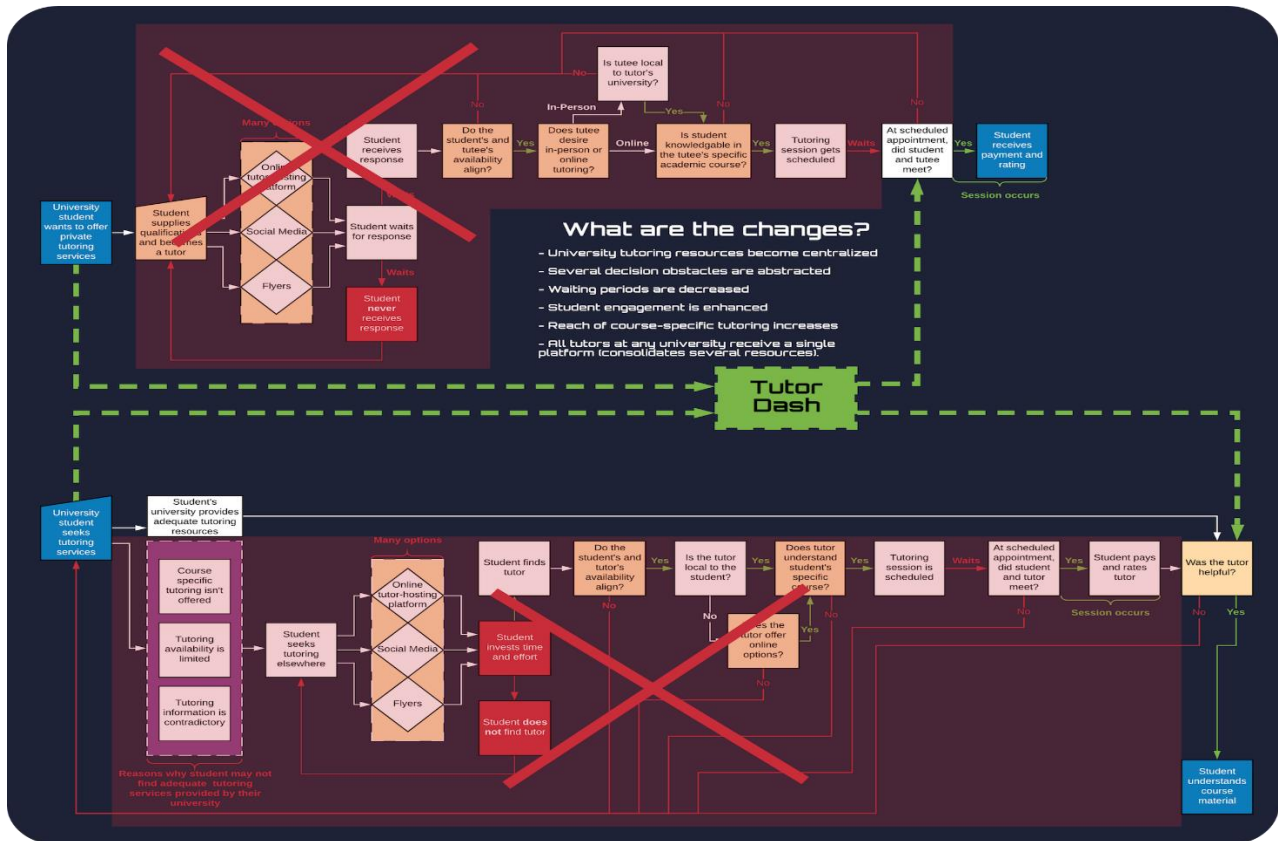
Figure 3: How Tutor Dash Affects the Current Processes

Tutor Dash merges the two current process flows illustrated in Figures 1.1 and 1.2. By providing a niche tool that connects students at the source, many of the current decision obstacles are bypassed.

Tutors and tutees must download the application, and then sign up for an account. Once membership has been established, users will then be notified about other potential users based on their login states (i.e., logged in as tutor or logged in as tutee).
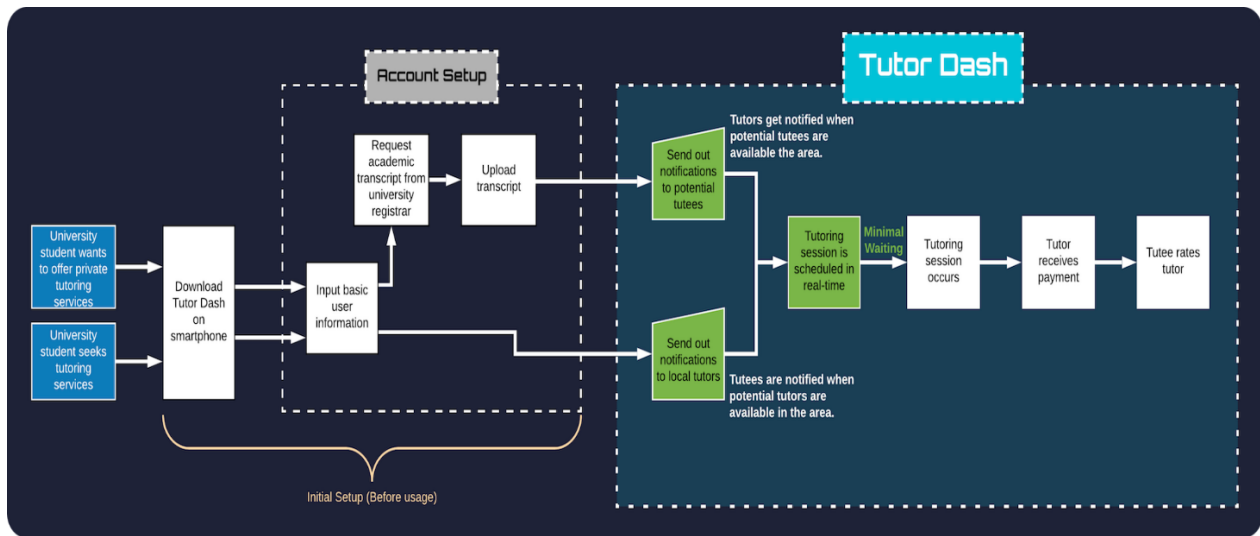
Figure 4: Tutor Dash Process Flow

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

Tutor Dash seeks to be a product that fills in gaps that currently exist within the private tutoring market, that centralizes the tutoring platform amongst university students, and that develops a customer base of regular users making use of the tutoring resources available to them. Additionally, it is Tutor Dash that is hoped to be able to resolve the issue and negative stigma of students seeking tutoring. A long-term goal for the project is to eventually expand this application such that it is used at multiple universities. The primary objective with implementing Tutor Dash is to increase convenience for college students when trying to make use of tutoring services.

2.2 Features and Capabilities

The Tutor Dash project is innovative through the consolidation of many prominent existing tools in such a way that it supports our objectives. These tools specifically allow us to constrain the user-base strictly to university students, qualify users as tutors in real-time based on their transcripts and previously taken courses, implement multiple rating systems for both tutors and tutees, allow any course at any given university to have tutoring services offered in-app, automatically calculate competitive pay-rates based on market parameters, allow for both in-person and online meetings, and provide a mechanism for automatic payments within the app.

Specifically, when creating a Tutor Dash account the user must supply a valid .edu email address marking them as a student at an American university; moreover, if the user chooses to apply to become a tutor, then the user must upload a transcript that verifies skill and proficiency in the courses that the user has taken and that they would like to tutor. The transcript that is uploaded is parsed by an algorithm that determines the user's eligibility to tutor on an individual course basis. Furthermore, the grades that a student earned in all previously taken courses at their university are analyzed, and if the student earned a B or higher in a normal course or a pass rating in a pass/fail course, then they are marked as being eligible to tutor that particular course if they so desire. After transcript parsing has fully completed the users will be presented with a list of all eligible courses to tutor, and from this list they will be allowed to prune it and select courses to include only those that they feel most comfortable assisting others with.

Additionally, in order to allow users more comfort when selecting a tutor, in Tutor Dash will be implemented a rating and review system. Tutees will be able to rate tutors who they have been helped by and write reviews for them after completion of a tutoring session. These ratings will be visible to all tutees who view tutor profiles when searching for someone to receive help

from. They will then naturally select one of the tutors who has a better rating, and this creates a

competitive environment for tutors to function in and that encourages them to provide genuinely

helpful assistance to students who need it.

Tutors will be able to set their availability as either "active" or "inactive". If a tutor user

marks itself as "active", then they will appear to tutee users on the live map of tutors and the

tutee will know that they are able to receive help from that tutor at the current moment if

necessary. If there are no available tutors for the subject that the tutee would like to review, then

the tutee can send out a request to all eligible tutors who can help with that particular course. The

tutors will see this request as a notification and can respond to it by either denying it or

scheduling a session with the tutee. If the tutee believes that a particular tutor has been beneficial

to the progress of their education, then they can choose to send a request to that specific tutor.

A tutee can initiate the scheduling of a session through the proposition of the means with

which the session will be held. That is to say that a tutee can request either a face-to-face

meeting or one that is held online. When scheduling a meeting in person, all both people have to

do is agree on the time and place when and where it is held. This communication between users

will occur through a peer-to-peer chatting feature of Tutor Dash so that all steps of the tutoring

sequence take place within the application.

Lastly, Braintree has a third-party library that will be incorporated into Tutor Dash to

handle the payments for tutoring services. Through the designation of master merchants and sub-

merchants in BrainTree's integration with our application, Tutor Dash itself is able to make

money through the collection of a small percentage of all transactions that take place within the

app. Tutors will receive the remaining amount of the tutoring session transaction after this

deduction has been made, and the tutor's hourly rate is calculated by an algorithm further

described in section 4.2.3.

2.3 Major Functional Components

Tutor Dash has major functional components consisting of the smartphone hardware on

which the application will run, the software, and end user entities. The users will download Tutor

Dash onto their phones and be given access to the functionalities of the application. Figure 5 is a

major functional component diagram that shows the full set of components in Tutor Dash and
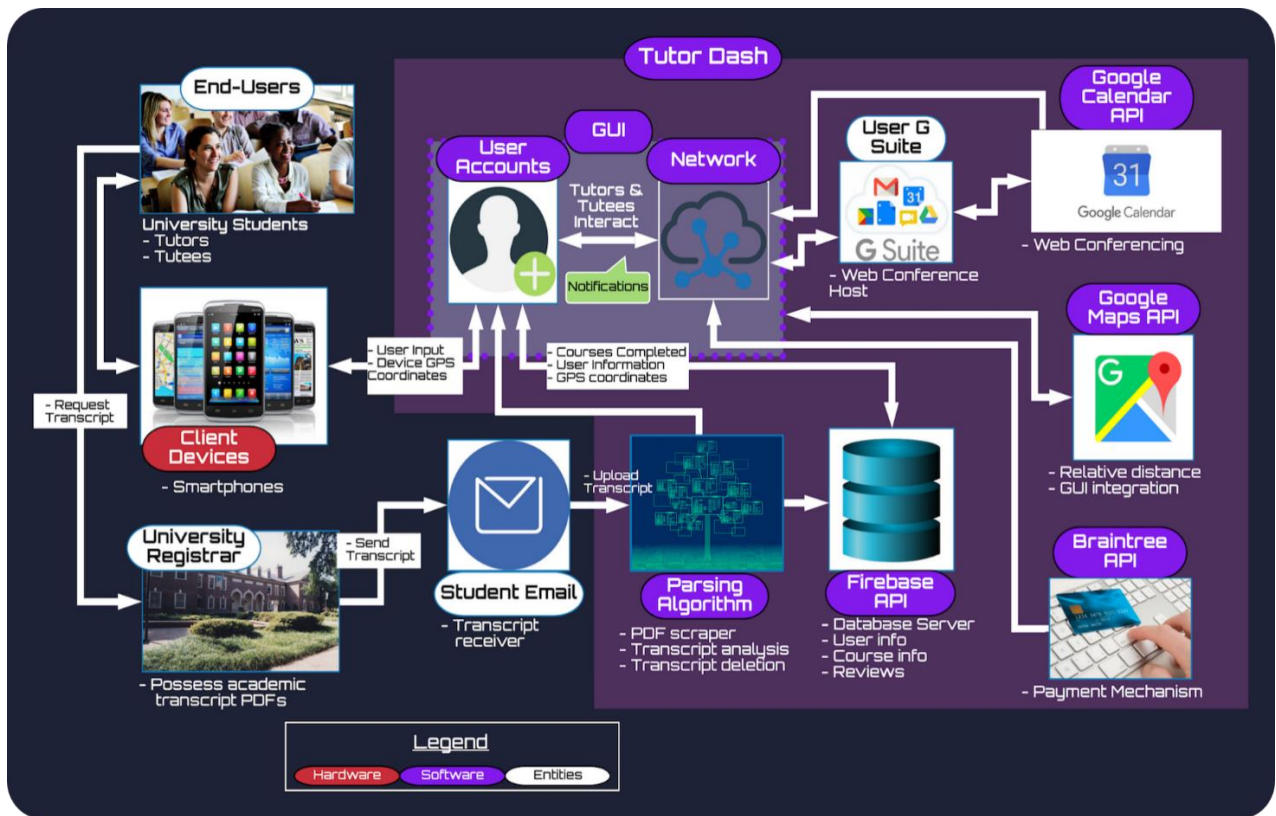
how they relate to one another.

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 for Tutor Dash will consist primarily of the graphical user interface, user accounts, transcript analyzation algorithm, and third-party applications like Firebase, Braintree, Google Maps and Google Calendar. The GUI is designed as a sequence of individual layouts for the various screens that make up the app interactions. As the Tutor Dash prototype will be run exclusively on Android mobile devices, the GUI and its layouts are developed in Android Studio through the XML filetype. A user's account must be created for one to use the application, and it consists of all associated user information like name, whether they are or are not a tutor, the courses they can tutor and their pay rate, their rating as a tutee and a tutor and the tutoring sessions they have scheduled and have completed. This account information will be stored in the Firebase database and authenticated through their authentication service.

2.3.2 Hardware

The hardware that a user must have for the Tutor Dash prototype to run is an Android mobile device.

2.3.3 Entities

The entities involved with the functionality of Tutor Dash include the tutor and tutee university students, a student email that they can register with, the university registrar that a user must request their transcript from should they choose to upload it to Tutor Dash and become a tutor, and a GSuite for use in scheduling of web-based tutoring sessions.

3. Identification of Case Study

For the prototype of the application, the end-user and customer bases for Tutor Dash will be restricted to only consist of students in attendance at Old Dominion University. These

students are people who would like to either provide help to fellow students through the offering

of private tutoring services or receive extra assistance with enhancing their grade and

understanding of ODU course content. The potential tutors at ODU may be looking to either

expand their existing private tutoring client base or make extra money by tutoring whenever is

convenient and they have time to do so.

Tutor Dash will be a centralized platform for all tutoring needs, and this can greatly

benefit students using the application as well as the university they attend. DFWI rates could

decrease because of the positive correlation between the availability of tutoring services and

student understanding of a course. If there is more readily available tutoring to be offered, then

students will be able to more thoroughly review the lessons of a course and as a result complete

the course with a better grade. This would greatly aid students in saving both their grades and

any tuition money that they would have lost if they did not receive tutoring and failed the course

as a result. Tutor Dash seeks to expand to include students on other college campuses.

[This space is intentionally left blank.]

4. Tutor Dash Prototype Description

The Tutor Dash prototype will not have the same features as the final real-world product.

Table 1 lists the similarities and differences between the two versions of the application.

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 Tutor Dash prototype will serve as a proof-of-concept for the real-world product. The core features of the application that will be implemented include the verification and qualification algorithms used when a user applies to become a tutor, the rating system, and the utilization of existing APIs to handle payment processing and tutoring sessions that are held through web conferencing. Essentially, Tutor Dash consolidates a set of existing tools with the addition of new features like pay-rate calculation and the automation of meeting setup and payment processing for greater user convenience and accessibility to the application.

4.1 Prototype Architecture

      The prototype architecture is comprised of all the components listed in the real-world product MFCD of section 2.3. As this is simply a prototype, only ODU will be used for its university registrar.

4.1.1 Prototype Hardware

      The only hardware necessary for the proper functionality of the Tutor Dash prototype is an Android smartphone that runs an updated version of the Android operating system. The RWP can be further developed through the implementation of an iOS version that can be used on iPhone smart devices.

4.1.2 Prototype Software

As was previously mentioned, user account information will be stored in the Firebase database and authenticated through the Firebase Authentication service. The number of queries that can be made to the database are limited per week, and storage size for user accounts can be upgraded at a later point for a RWP that requires more space.

4.2 Critical Design Components

Tutor Dash as a mobile application consists of three major components that work in unison to fulfill all proposed functionalities.

4.2.1 Database

The specific Firebase database used in Tutor Dash is Cloud Firestore. This is a NoSQL, JSON, document-based database that provides cloud storage, easy scalability to our application and its number of users. As Firebase is owned by Google, the APIs have been designed in such a way that they are well-suited for Android apps and are easy for the team to implement and incorporate into the project. The database will store all user and school data as well as the records of payments, reviews, user schedules, and chat histories between users. JavaScript Object Notation (JSON) is used to store, access, and update JavaScript objects that are converted into JSON text for the purposes of communicating with the server. The following Table 2 shows specific depictions of the schema.

[This space is intentionally left blank.]

User

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

School

schoolID
schoolName
schoolSuffix
courses

Courses

courseID
courseName
meanPayRate
stdDev

Chat

UID1_UID2
senderName
sendeeName
message
timestamp

Blacklist

email
schoolID

Reviews

UID
reviewerUID
rating
comment
timestamp

Payments

UID
receiverUID
dateTime
amount

Schedule

schoolID_UID
date
eventID
eventName
startTime
stopTime

Table 2. Database Schema

4.2.2 UI/UX

The proper design and implementation of Tutor Dash's user interface and user experience are crucial to the successfully accumulation of users. Figure 6 displays the sum total of all phases and views of the UI layouts and how the user will typically progress through them as they use Tutor Dash.

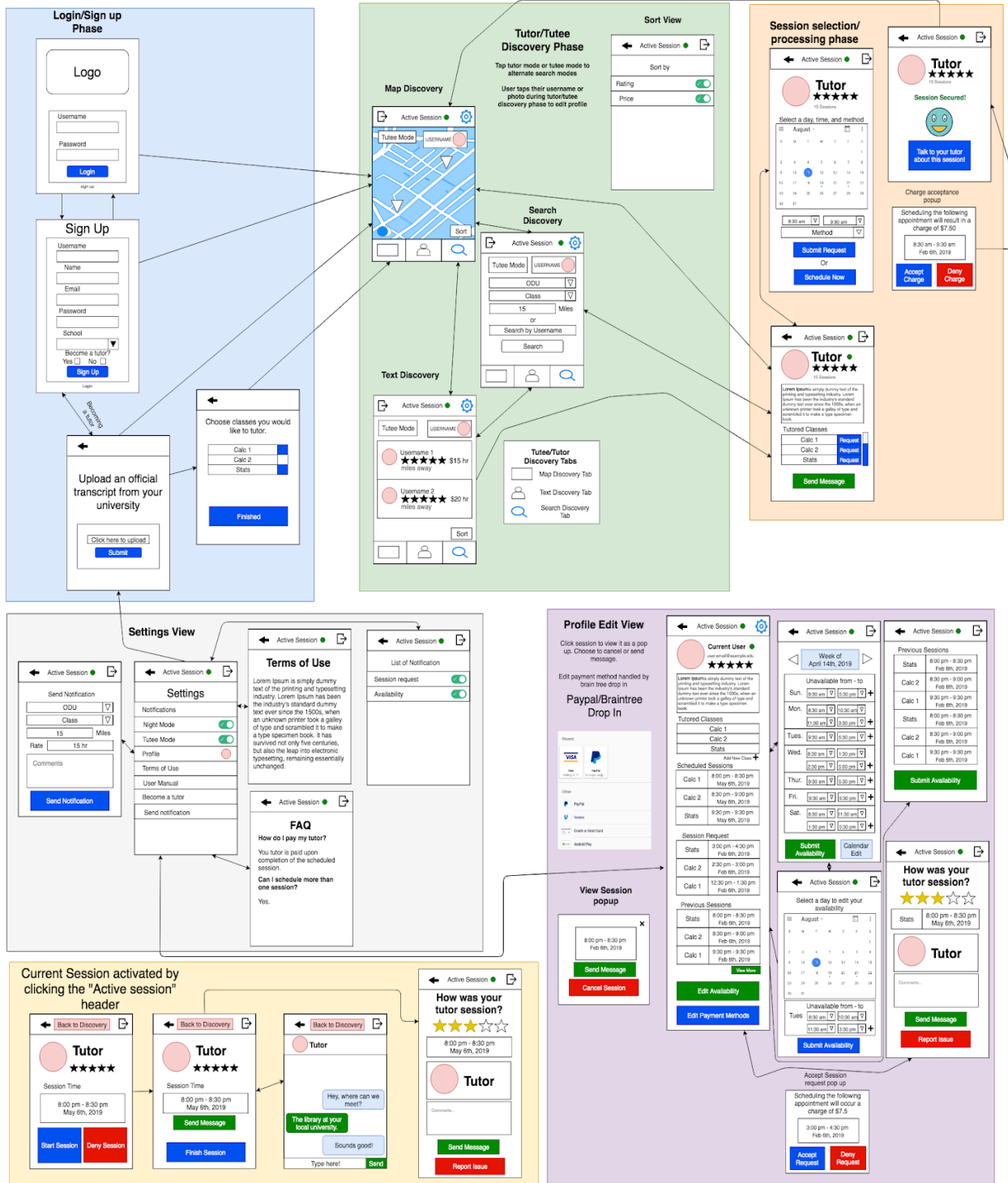[This space is intentionally left blank.]

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.

Views can be selected at any time from most of the phase layout screens. Figures 7 through 11 all show a more detailed and segregated version of the individual phases of the UI. All of the mentioned phases are comprised of multiple UI layouts that are required for the execution of a specific activity.
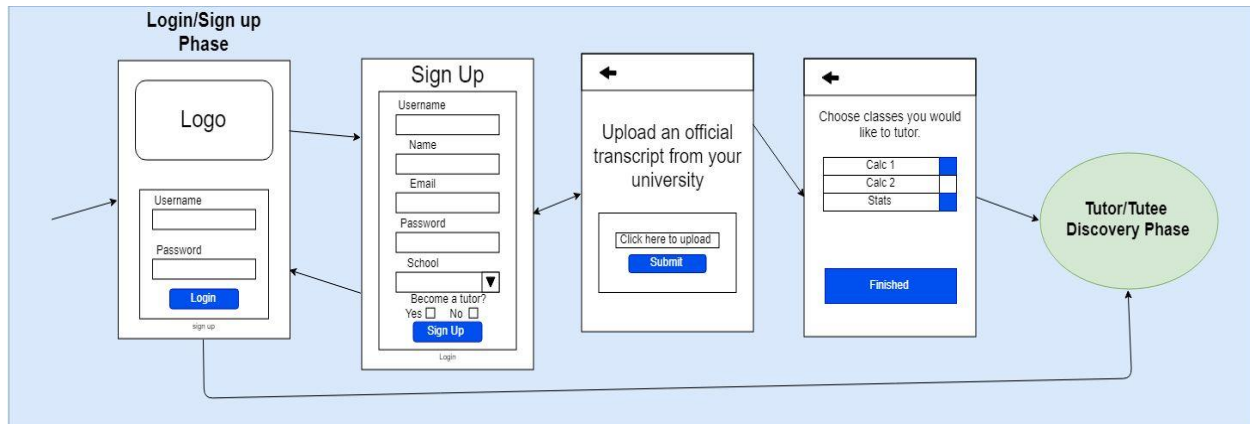
Figure 7. UI/UX Phase 1

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 the are signing up and wish to become a tutor, then they can provide their transcript during this phase if desired.

[This space is intentionally left blank.]
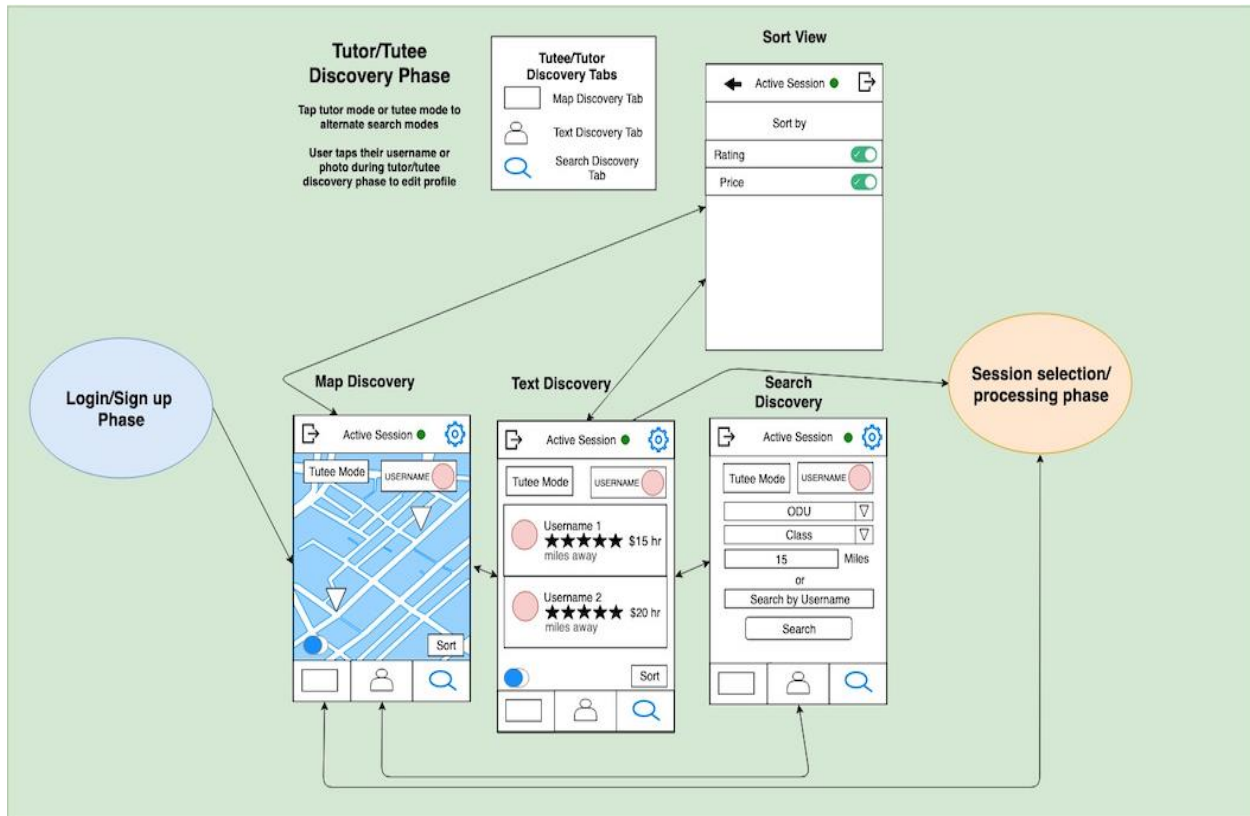
Figure 8. UI/UX Phase 2

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
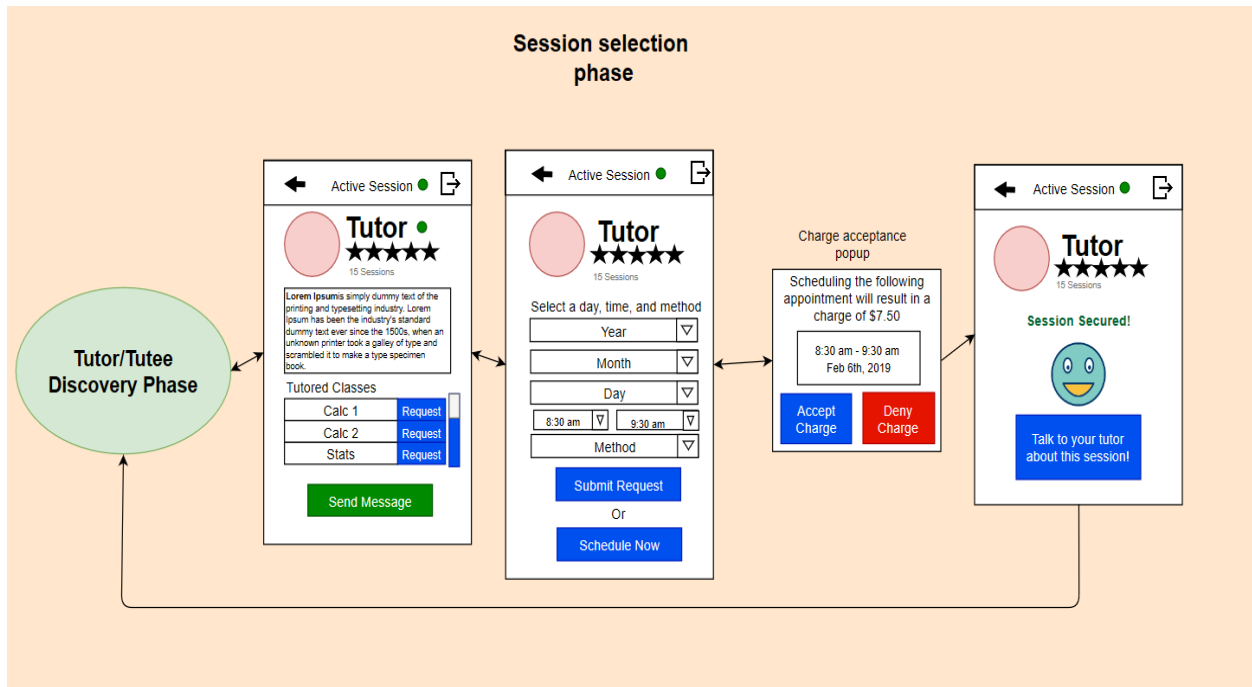
parameters.

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.
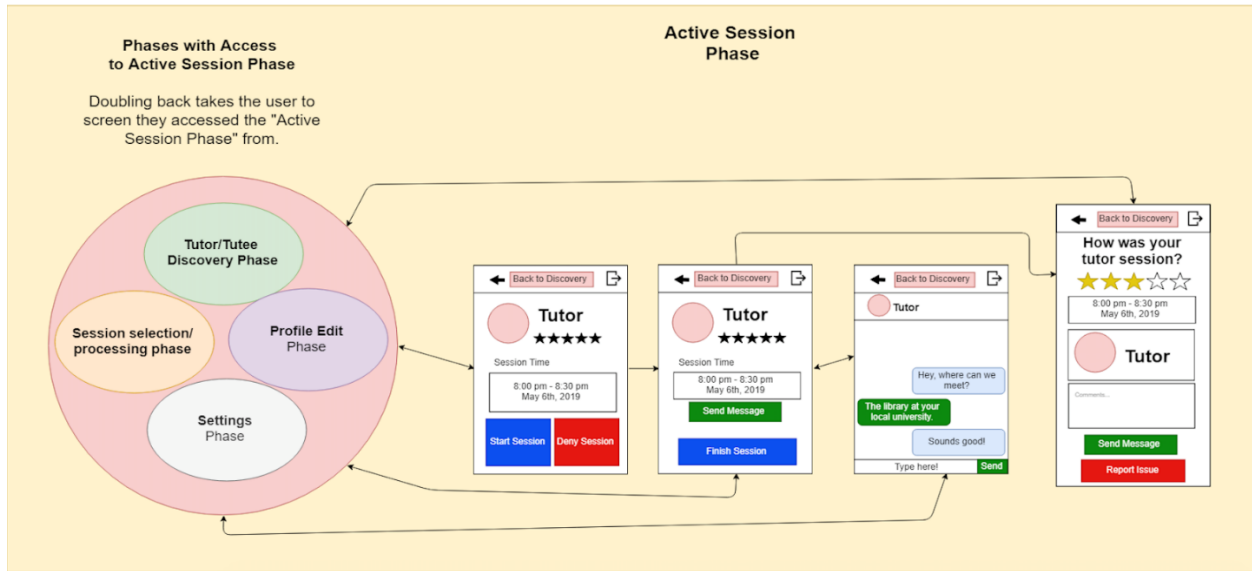
[This space is intentionally left blank.]

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.
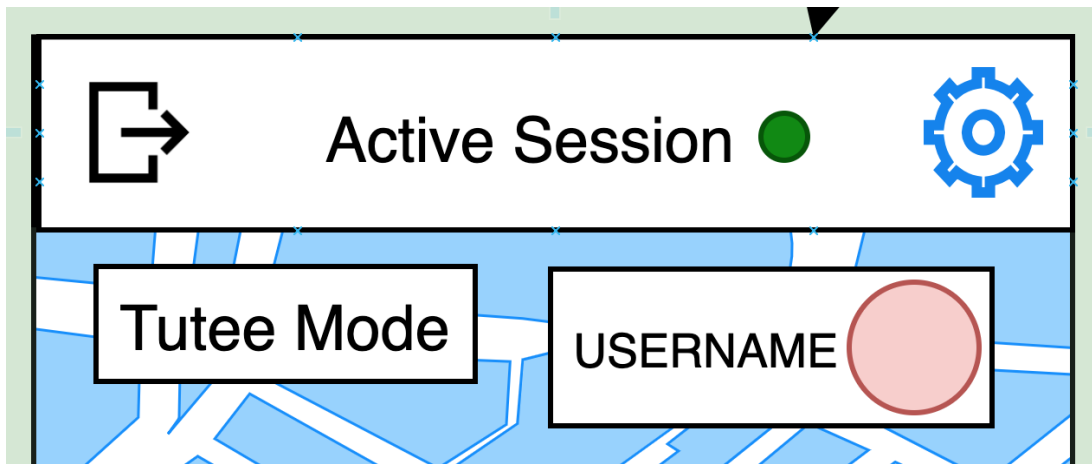

Figure 11. UI/UX Phase 4B

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.

Figures 12 and 13 show both of the Tutor Dash views. Users can access views from any

phase of the application, and from the view layouts they can observe and update their settings
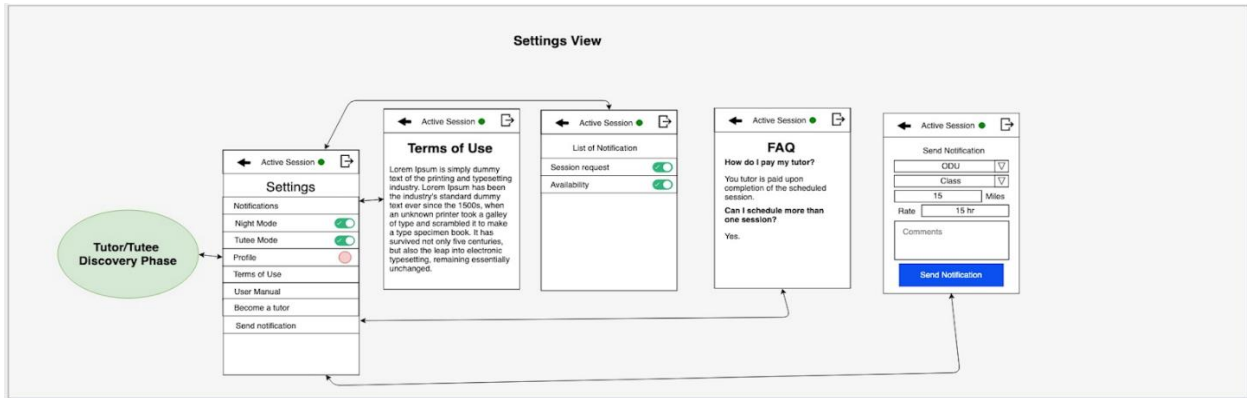
and user profile.

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.

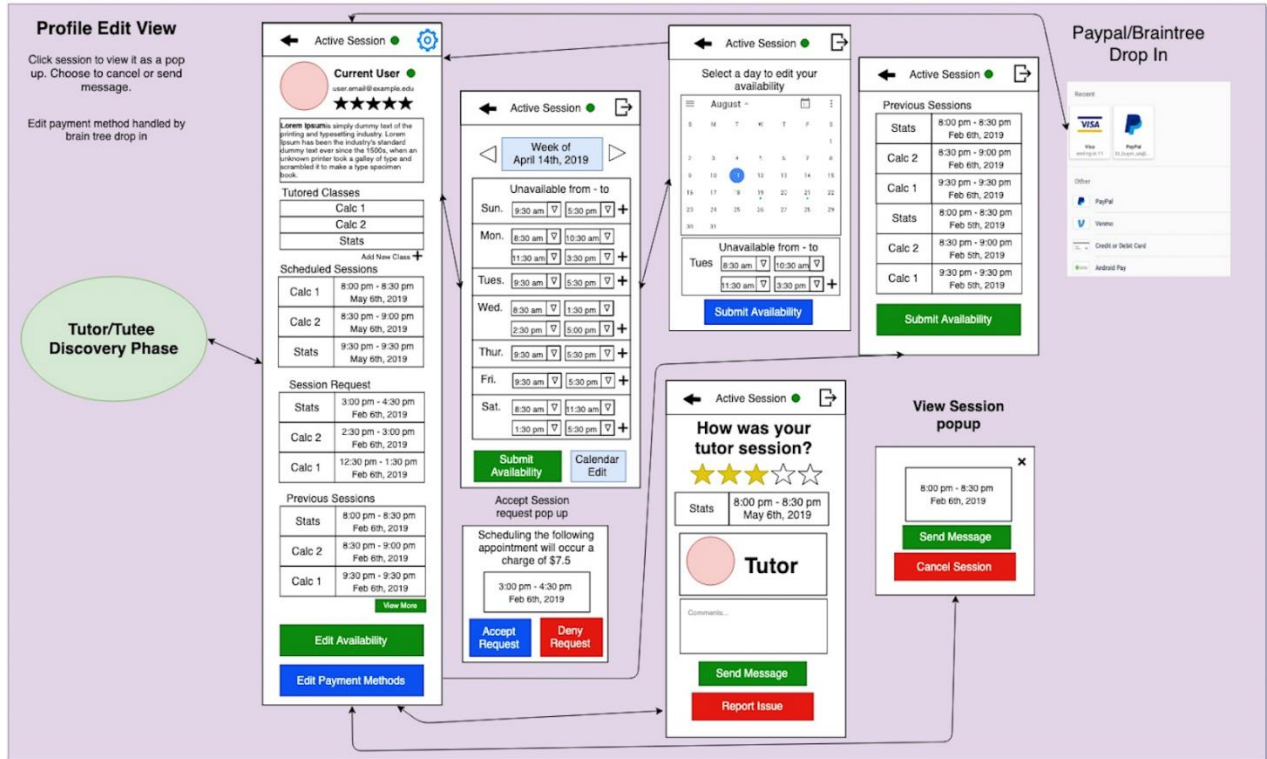[This space is intentionally left blank.]

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 algorithms displayed in Figures 14 through 18 show the primary backbone

functionalities required for the proper execution of Tutor Dash. These algorithms include the

transcript parser, relative distance calculator, competitive pay-rate calculation algorithm, web-

conference event creator, and payment algorithm.
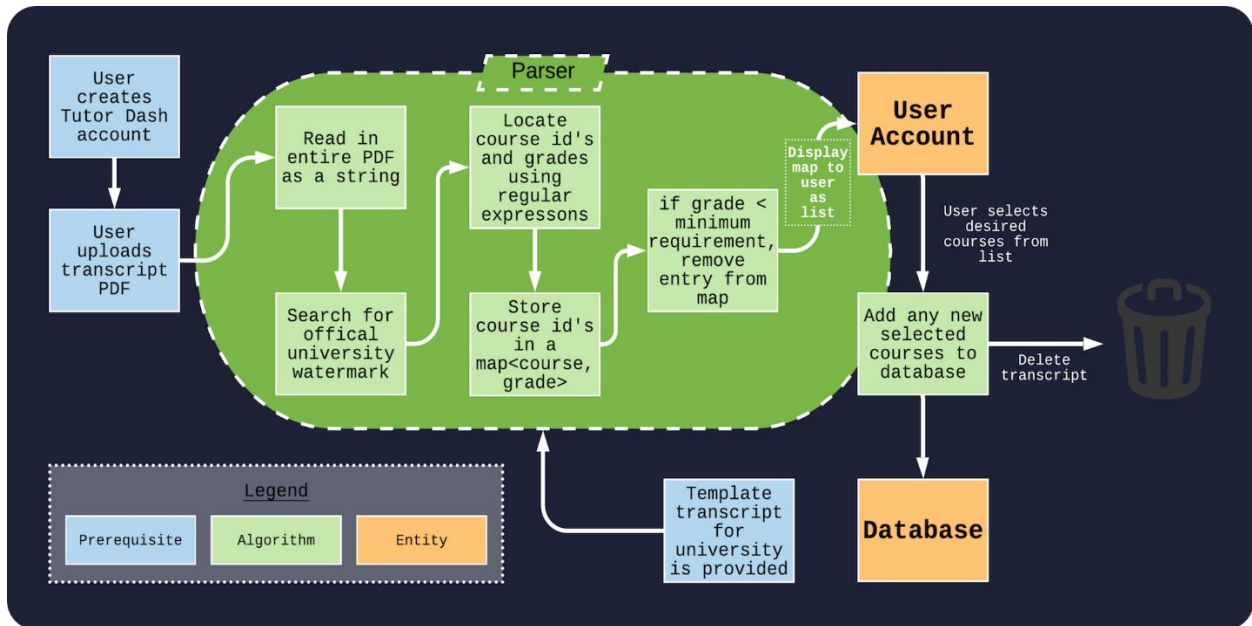

[This space is intentionally left blank.]

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.
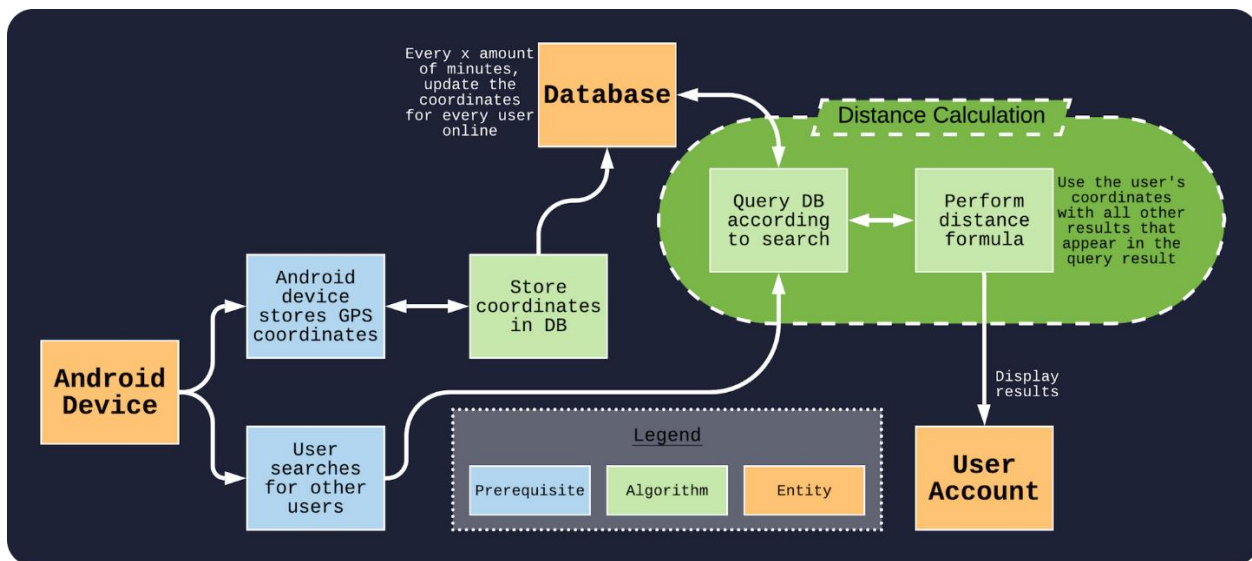

Figure 15. Relative Distance Calculator Algorithm Flow

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.
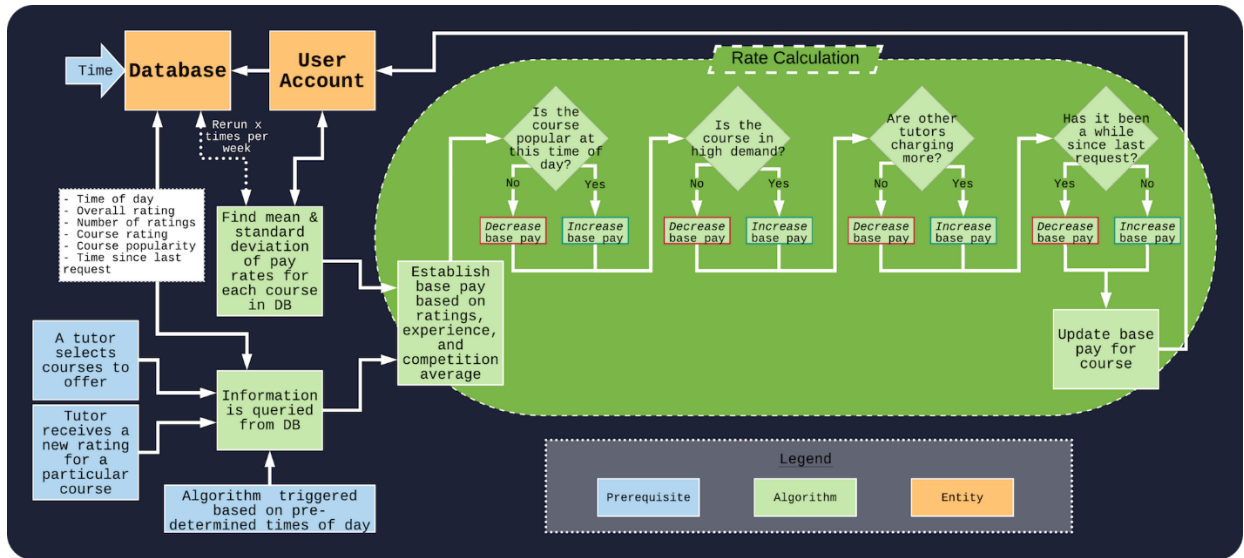

Figure 16. Competitive Pay-Rate Algorithm Flow

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.
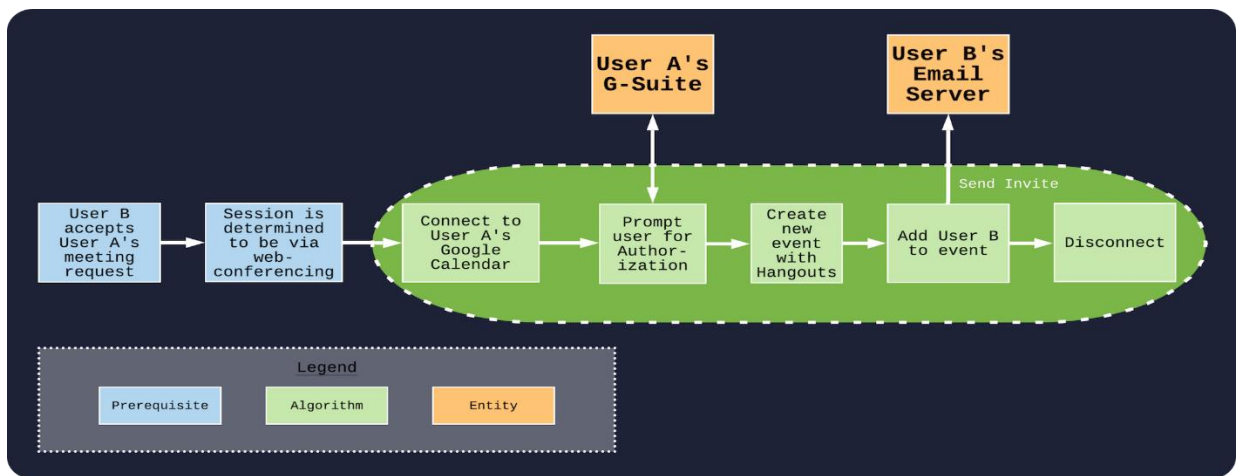

Figure 17. Web-Conference Appointment Creator Algorithm Flow

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.
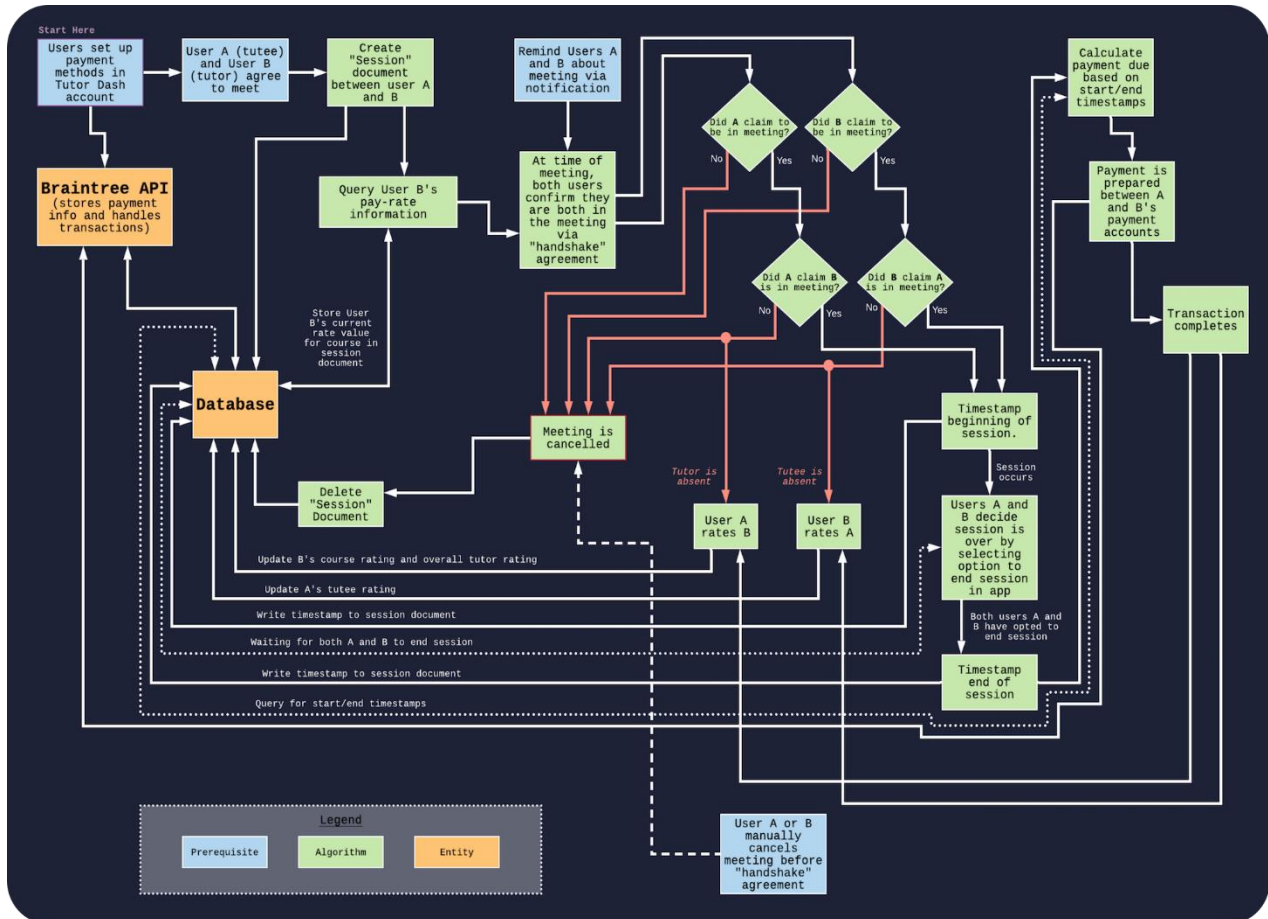

Figure 18. Payment Logic Algorithm Flow

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.

[This space is intentionally left blank.]

4.3 Prototype Features and Capabilities

The prototype will demonstrate a user-friendly application that allows tutee customers to connect to their respective tutor counterpart. The payment method used in tutoring session transactions is a Braintree plugin that allows for simplicity in the payment process. The prototype will demonstrate an intuitive user interface, a transcript input parser to prove a potential tutor's knowledge base, a connection to Google's Hangout meeting and map systems, a user rating system, and a chat window for both tutors and tutees to communicate.

The objective of Tutor Dash is to create a centralized, understandable platform so that there are no frictions when an individual is looking for tutoring help. Because the application will have an intuitive design, simple tutor/tutee interactions and finder, and a method of spontaneous payment and meetup, then the objective will be achieved.

Tutor Dash will demonstrate success through the display of an application prototype with minimal product bugs. The developed prototype will inherently solve the problem defined by the case study by creating a centralized environment where tutors and tutees can provide or seek out tutoring services for myriad courses at their university. As was previously mentioned with the introduction of Table 1, there will be differences between the RWP and the prototype. Features including night mode, user reporting and blacklisting, refunds, and free sessions or bonus monetary incentives will either not be implemented or fully implemented in the prototype.

4.4 Risks and Mitigations

One can find the potential risks for the Tutor Dash project in Figure 19, where it describes the possibility of each risk presenting an issue and how severe the impact of this issue would be on Tutor Dash's functionality. Each risk in this figure is associated with a specific label and definition that can be located in Figure 20.
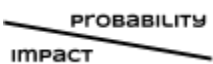
PROBABILITY / IMPACT	Very Low	Low	Moderate	High	Very High
Very High	T3, T4, L1	T6, L2		C3, C4	
High	T1, C7	C6, T9			
Moderate	C5	T8	C1, C8, T7		
Low	C2, T5		C10, T2		
Very Low		C9			

Figure 19. Risk Matrix Overview

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.

Figure 20 defines what the risks are and how the Tutor Dash team plans to mitigate them through a number of software development strategies.

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 Description

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 primary objectives of the development of Tutor Dash include the design of a clean and easily navigable user interface that aids in increasing user retention, the ability for a user to find a tutor or tutee in a timely manner, a functioning payment system, a stable connection to a medium through which the tutoring session is held, and a rating system that cannot be abused.

4.6 Prototype Development Challenges

The development team of Tutor Dash faces a number of challenges. One such challenge is the fact that many of Team Gold's members are inexperienced in Android applications development and have not worked with the third-party APIs mentioned in the MFCD. The implementation of web conferencing and payment processing will be difficult because of permissions issues dealing with G Suites and monetary values. The connection of the user interface to the backend algorithms and database will be difficult because of the clear division of work on UI/UX and databases and because some developers may not always be available to work on their portion of the project. The sending and receiving of messages between users in real time, the handling of funds between tutees and tutors, the implementation of a handshake agreement to initialize the start of a meeting, the sending of only a reasonable amount of alert notifications regarding compatible tutors and tutees to users, the assignment to tutors of an accurately calculated competitive payment rate dependent on multiple factors, the testing of the user experience, and the development time constraints on the project are all projected challenges for Team Gold.

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 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.

6. 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_au

   thors_&utm_medium=Email&utm_content=T1_11.5e1_author&utm_campaign=Email_publicati

   on&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#A

N=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