

Lab 1 – Tutor Dash Description

Dwight C. Owings

CS 411

Professor T. Kennedy

12 September 2019

Version 1

Table of Contents

1. Introduction..... 5

2. Tutor Dash Product Description 7

2.1 Objective and Goals of Tutor Dash 9

2.2 Features and Capabilities 9

2.2 Major Functional Components 10

2.3.1 Entities 11

2.3.2 Hardware..... 11

2.3.3 Software..... 12

3. Identification of Case Study..... 12

4. Tutor Dash Prototype Description 13

4.1 Prototype Architecture 14

4.1.1 Prototype Hardware 14

4.1.2. Prototype Software..... 14

4.2 Critical Design Components..... 15

4.2.1 Database..... 15

4.2.2 UI/UX 17

4.2.3 Algorithms 24

4.3 Prototype Features and Capabilities..... 28

4.4 Risks and Mitigations 29

5 Glossary 32

6 References..... 34

Lab 1 – Tutor Dash Description	3
--------------------------------	---

List of Figures

Figure 1: Students Seeking Academic Assistance.....	6
Figure 2: Students Seeking Tutoring Advertisement.....	6
Figure 3: How Tutor Dash Affects the Current Process Flows.....	8
Figure 4: Tutor Dash Process Flow.....	9
Figure 5: Major Functional Component Diagram.....	11
Figure 6: UI/UX Full Implementation.....	18
Figure 7: UI/UX Phase 1.....	19
Figure 8: UI/UX Phase 2.....	20
Figure 9: UI/UX Phase 3.....	21
Figure 10: UI/UX Phase 4A.....	22
Figure 11: UI/UX Phase 4B.....	22
Figure 12: UI/UX View 1 – Settings.....	23
Figure 13: UI/UX View 2 – User Profile.....	24
Figure 14: PDF Transcript Parser Algorithm Flow.....	25
Figure 15: Relative Distance Calculator Algorithm Flow.....	25
Figure 16: Competitive Pay-Rate Algorithm Flow.....	26
Figure 17: Web Conference Appointment Creator Algorithm Flow.....	27
Figure 18: Payment Logic Algorithm Flow.....	28
Figure 19: Risk Matrix Overview.....	29
Figure 20: Risk Matrix Descriptions.....	30

List of Tables

Table 1: RWP vs Prototype.....13

Table 2: Database Schema.....16

1. Introduction

According to Hanover Research (2014), many universities offer official tutoring services, but they tend to have several major issues, including poor advertising and schedules that do not work for all students. Despite these problems, tutors offer many benefits to universities, including a decrease in DFWI rates and increase in student retention. One of the major problems faced by many universities' tutoring services is a lack of organization. Students often find it difficult to use these resources, and qualified students are discouraged from becoming tutors themselves due to inconvenient payment and scheduling options.

Current tutoring resources, either on-campus or off-campus, do not suffice for many students. The lack of centralization in many university tutoring services makes it hard for students to use or understand official resources. Furthermore, many university students would become tutors if it were easier, but no centralized platform currently exists that offers them that opportunity.

Figure 1 shows the current process flow for tutors seeking tutees. The tutor must wait for an uncertain amount of time before the process starts, and the process could stop at any point for several factors outside of their control.

[This space intentionally left blank]

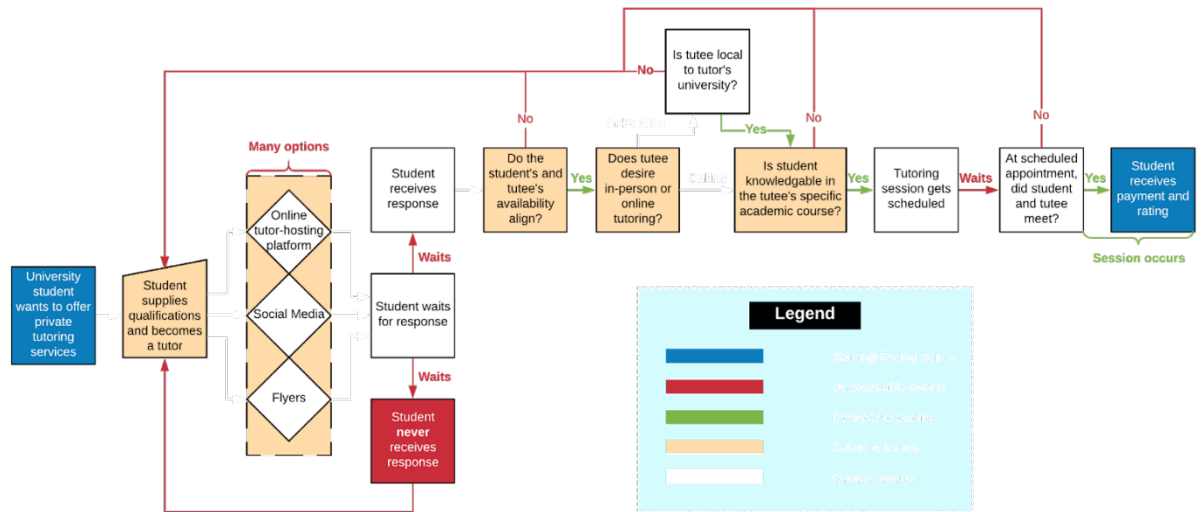


Figure 1: Students Seeking Academic Assistance

Figure 2 shows the current process flow for tutees seeking tutors. Similarly to Figure 1, tutees must spend time searching for a suitable tutor, and the process could end before one is found for reasons such as incompatible schedules or an inability to find a location for a meeting.

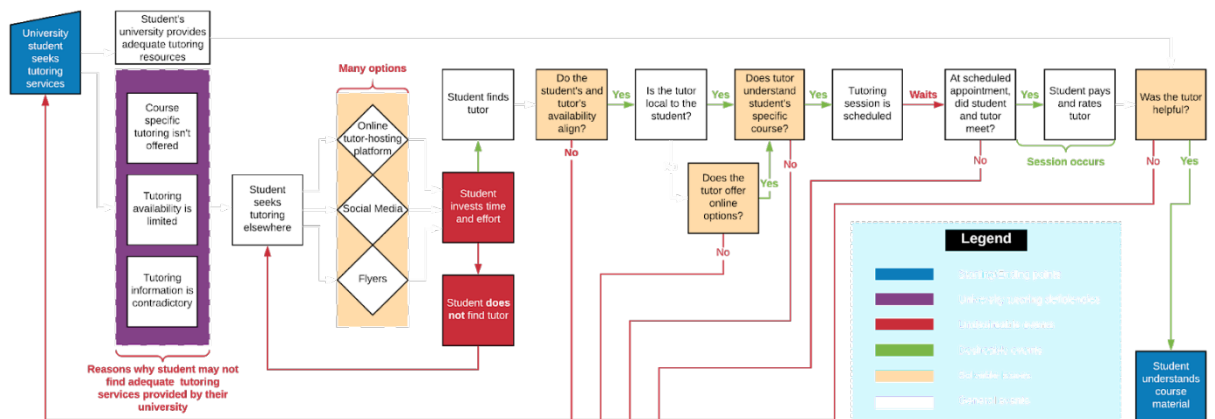


Figure 2: Students Seeking Tutoring Advertisement

Tutor Dash aims to solve these problems by acting as a centralized resource for university-specific tutoring. With convenient scheduling and course-specific tutoring, Tutor Dash can ensure that all students have an opportunity to reach out for help if they need it.

2. Tutor Dash Product Description

Both current process flows are flawed, but Tutor Dash will correct these flaws by combining the process flows together and eliminating the issues they contain, as shown in Figure 3.

[This space intentionally left blank]

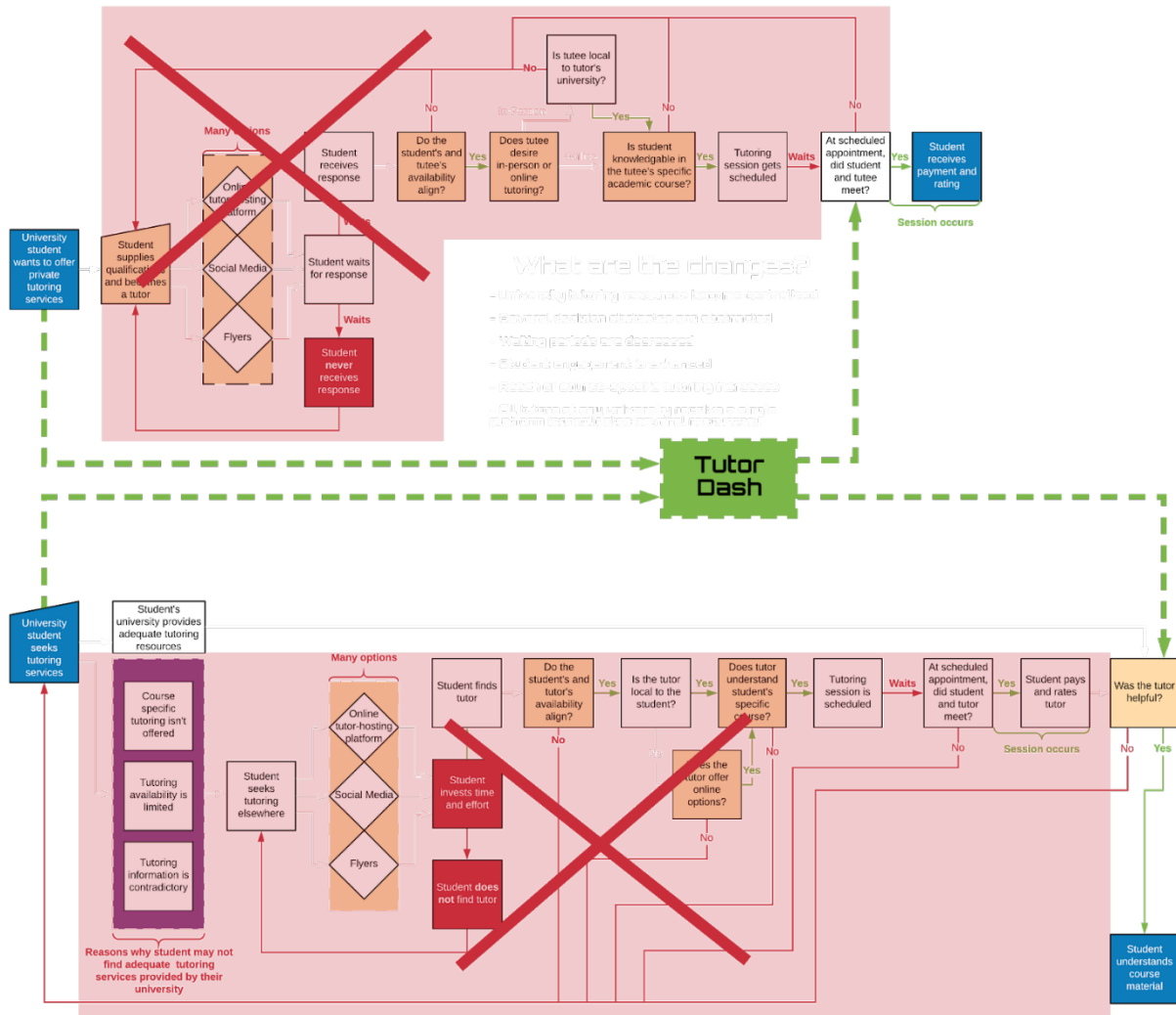


Figure 3: How Tutor Dash Affects the Current Process Flows

As shown in Figure 4, Tutor Dash will reduce time spent waiting for a response by optionally notifying tutors when a tutee looking for their services comes within a certain distance, and doing the same for a tutee upon detecting a tutor offering to teach a subject that they wish to learn. Furthermore, once both parties have agreed to a tutoring session, Tutor Dash will schedule the session in real-time, either in person or online, so that both parties can benefit from the meeting with minimal time spent waiting.

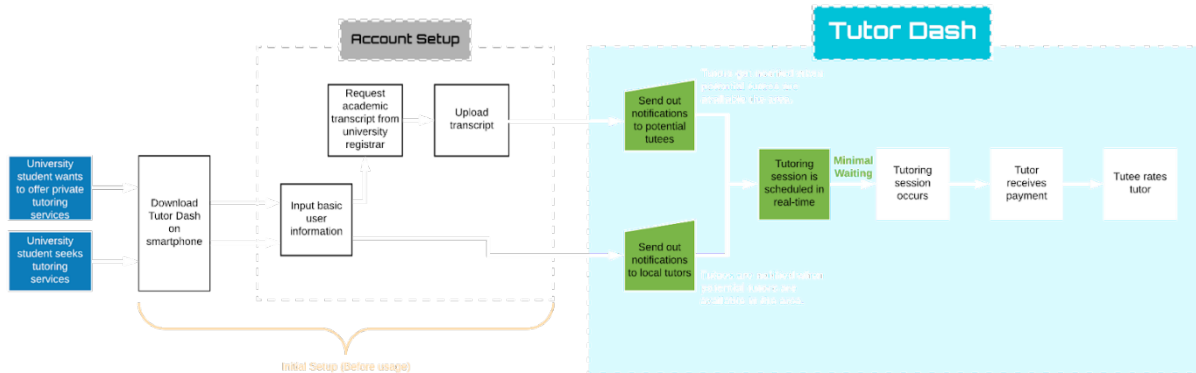


Figure 4: Tutor Dash Process Flow

2.1 Objective and Goals of Tutor Dash

The main objective of Tutor Dash is to create a product that fills in the gaps that exist in the current private tutoring market. Furthermore, Tutor Dash will serve as a centralized tutoring platform specifically designed for university students. The prototype will only apply to Old Dominion University, but Tutor Dash is expected to expand to multiple universities and establish a base of repeat customers at several universities. The end goal of this product is to allow more university students to have better access to tutoring services, and to reduce the stigma associated with seeking a tutor.

2.2 Features and Capabilities

In order to become a tutor, a student must enter an official academic transcript, and will only be eligible to tutor courses that they have already taken and passed with at least a set minimum grade. This feature prevents students from signing up as tutors for courses that they do not understand thoroughly, and assures tutees that the tutors they are meeting with are knowledgeable in a given subject.

Tutor Dash will implement a meaningful rating system for both tutors and tutees. This system allows tutees to determine if a potential tutor is capable of teaching effectively, and allows tutors to determine if a potential tutee will attend a scheduled meeting. The tutor ratings will be split into a rating for each class they teach and an overall rating that averages all their class ratings. In order to ensure a fair price for both tutors and tutees, Tutor Dash will calculate rates for tutors based on experience, ratings, and other factors.

Tutor Dash will estimate the locations of its users that choose to use the feature, and can optionally notify tutees when a potential compatible tutor comes within a distance set by the user. This feature also applies to tutors seeking tutees. Tutor Dash will allow its users to set a schedule when they are considered available, which will determine whether they are considered as a potential tutor or tutee at a given time.

Tutoring sessions can be either online or in-person, and online meetings will be automatically setup through Google Hangouts when scheduling occurs. Payment for these sessions is handled in-app.

2.2 Major Functional Components

The hardware, software, and entities used by Tutor Dash are displayed in Figure 5. In addition to its own software, Tutor Dash will use APIs from third party companies, including Google, Firebase, and Braintree.

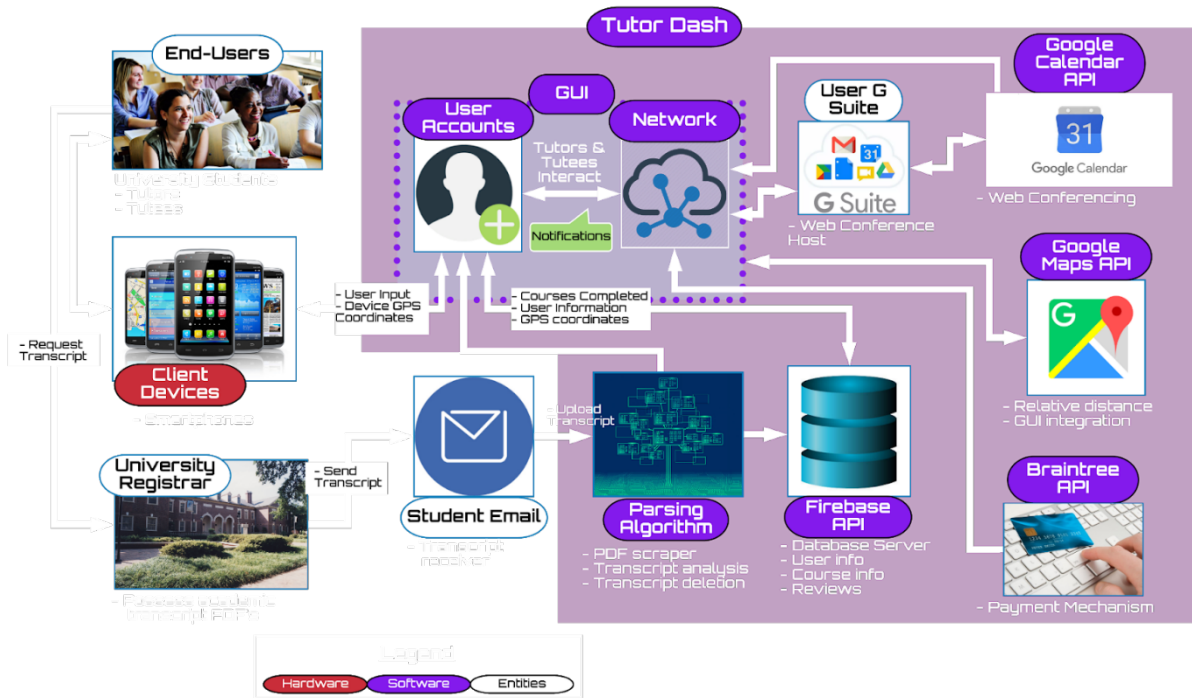


Figure 5: Major Functional Component Diagram

2.3.1 Entities

The most important entities of Tutor Dash are the end users, who are classified as tutors or tutees, depending on what they choose to identify as at a given moment. Students seeking to become tutors must request an official transcript from their university’s registrar and upload it to Tutor Dash. This process requires the use of a student e-mail account. Both tutors and tutees must use their G Suite, specifically Google Hangouts, in order to hold an online session.

2.3.2 Hardware

Tutor Dash will be a mobile application, so end users must have access to an Android smartphone to use it.

2.3.3 Software

In addition to the GUI, user accounts, and network contained within the application, Tutor Dash will use Firebase as a database and create a parsing algorithm used to gather relevant information from submitted transcripts. In addition, the application will use the Google Calendar API to set student schedules, the Google Maps API to track distances between potential tutors and potential tutees, and the Braintree API to handle in-app payments.

3. Identification of Case Study

Tutor Dash's current projected customers are ODU students seeking tutoring and ODU students seeking to tutor other students. Students are expected to seek to tutor others if they are already private tutors looking to expand their client base, or simply students that want to make extra money and have the knowledge and time to tutor their peers.

A centralized, easy to use platform for all tutoring needs would benefit both the students that use the service and the university itself. By seeking help when they need it, students could pass a class that they might fail otherwise, saving them money and keeping their GPA from falling. Because fewer students fail classes, the university's DFWI rates would decrease. Tutor Dash is expected to eventually expand its customer base to students at universities other than ODU.

4. Tutor Dash Prototype Description

As shown in Table 1, the Tutor Dash prototype will differ from the real-world product.

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

Blacklisting of users	Fully-Functional	Partially Functional - Capabilities will be implemented but not used
Support of multiple universities	Fully-Functional	Eliminated
Cross-platform support	Eliminated	Eliminated
Firebase console linked to test suite(s) with mockups	Eliminated	Fully-Functional

Table 1: RWP vs Prototype

The prototype will use mocked-up data that will not exist in the real-world product. Blacklisting of users, free sessions for loyal customers, and support for multiple universities will exist in the real-world product, but not the prototype. The automated pay-rate calculation that will exist in the real-world product will need to use mocked-up data in the prototype because data used to make that calculation will not exist in the prototype. Similarly, refunds will not yet be automated, and the report button will serve no purpose in the prototype. Core features of the application, such as the verification of potential tutors, the rating system, and the usage of third-party APIs will be implemented in the prototype.

4.1 Prototype Architecture

The prototype architecture is similar to what is shown in Figure 2, except the university registrar only refers to ODU.

4.1.1 Prototype Hardware

Tutor Dash will be a mobile application, so users must have access to an Android smartphone to use it.

4.1.2. Prototype Software

The Tutor Dash prototype will use Firebase as its database, and it will hold all data necessary for major functions. Users must have Android API 16 or greater to use Tutor Dash. Braintree will be used to process payment for both tutors and tutees.

In addition to third-party APIs, Tutor Dash will use several algorithms designed specifically for the application. A PDF transcript parser will be required to check the qualifications of potential tutors, a relative distance calculator will be required to notify users when a potential tutor or tutee enters a set radius, a pay-rate calculator will be required to set fair, affordable rates for tutoring sessions, a web conference creator will be required to quickly set up online meetings, and payment logic to determine how and when payment should occur.

4.2 Critical Design Components

The three main components of Tutor Dash will be its database, its UI/UX, and its algorithms. The database will be used as a server, the UI/UX will serve as the front-end and will be used to deliver content clearly to the users, and the algorithms will serve as the back-end and will handle most of the application's functionality.

4.2.1 Database

Firestore will serve as Tutor Dash's database. Table 2 describes the schemas.

describe payments made in-app. Finally, the blacklist schema will describe the users that have been blacklisted from the app.

4.2.2 UI/UX

Tutor Dash will have four phases and two views, as shown in Figure 6.

[This space intentionally left blank]

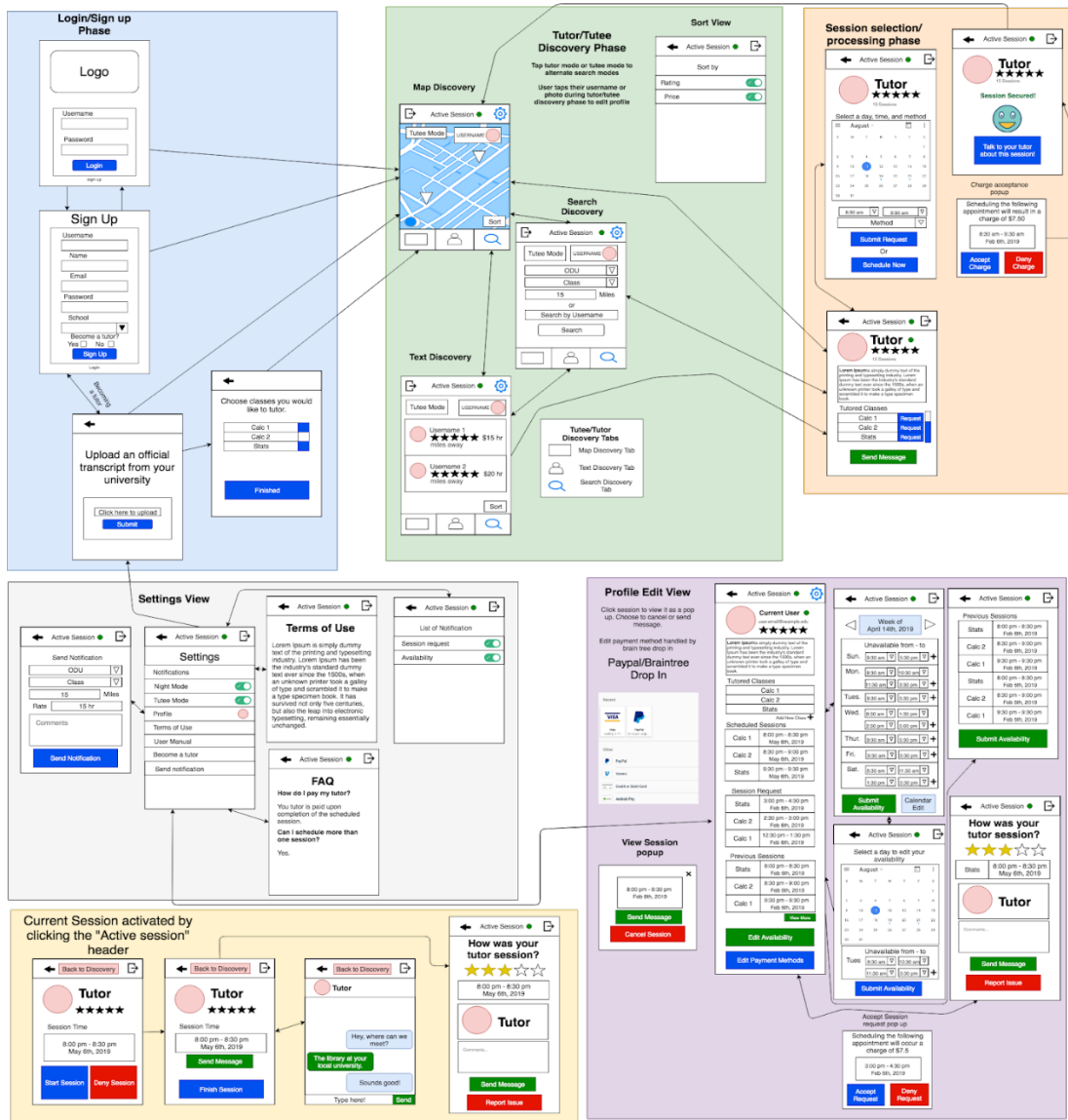


Figure 6: UI/UX Full Implementation

The Login/Signup phase, as shown in Figure 7, will allow users to log in to an existing Tutor Dash account or create a new account if they do not already have one.

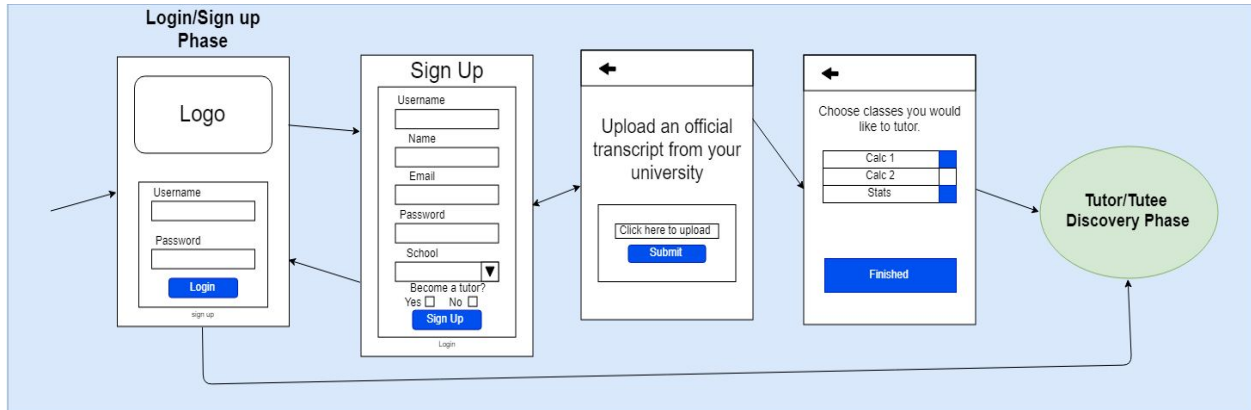


Figure 7: UI/UX Phase 1

From the Login/Signup phase, the user can navigate to the Tutor/Tutee Discovery phase. As shown in Figure 8, in this phase, users can switch between tutor and tutee mode and search for a match using a map of nearby users or a list of potential tutors and tutees. Alternatively, the user can search for a specific person by entering their username. These results can be sorted by rating or price.

[This space intentionally left blank]

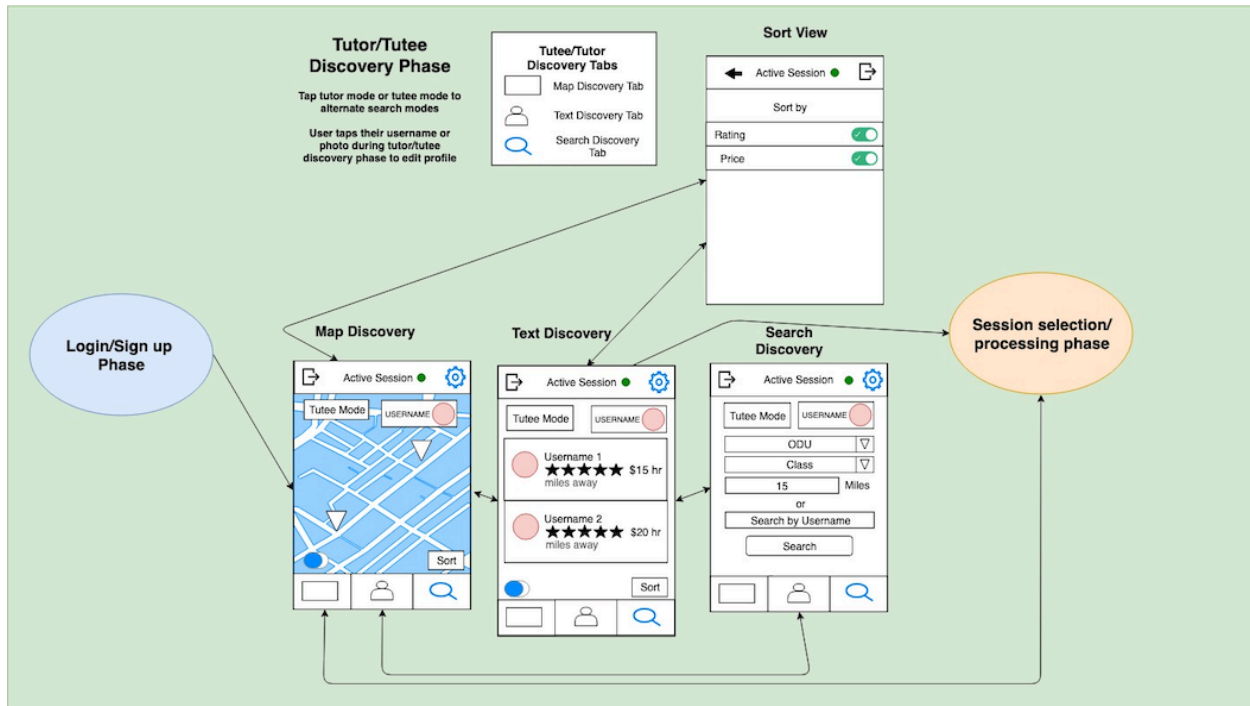


Figure 8: UI/UX Phase 2

Once a tutor has been selected, the tutee will be taken to the Session Selection phase, as shown in Figure 9. In this phase, the user will be able to send messages to their match and view their match’s username, ratings, and biography. The user will also be able to see a list of classes offered by the tutor they matched with. In this view, a user can either schedule a session during the tutor’s available times, or schedule one in real-time.

[This space intentionally left blank]

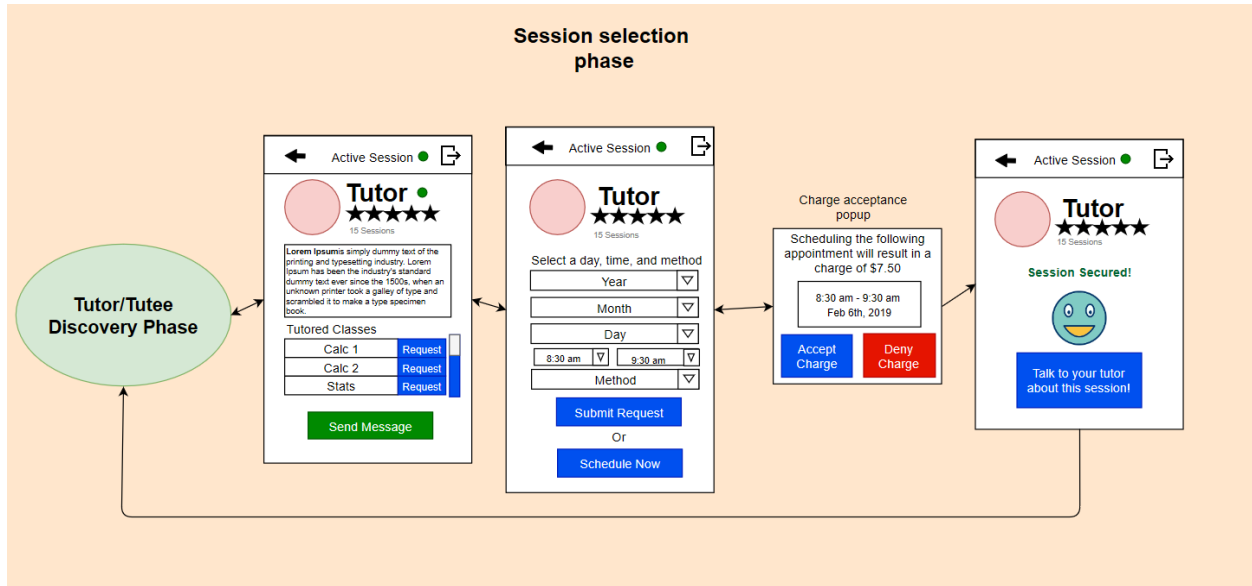


Figure 9: UI/UX Phase 3

The Active Session phase will be able to be accessed from most other phases in the application, as shown in Figure 10. In this phase, the tutee will be able to communicate with the tutor via an in-app messaging system, and the tutoring session is officially started and finished in this phase. Upon completion of the session, the tutor and tutee will rate each other, and, if necessary, report issues with the session, tutor, or tutee.

[This space intentionally left blank]

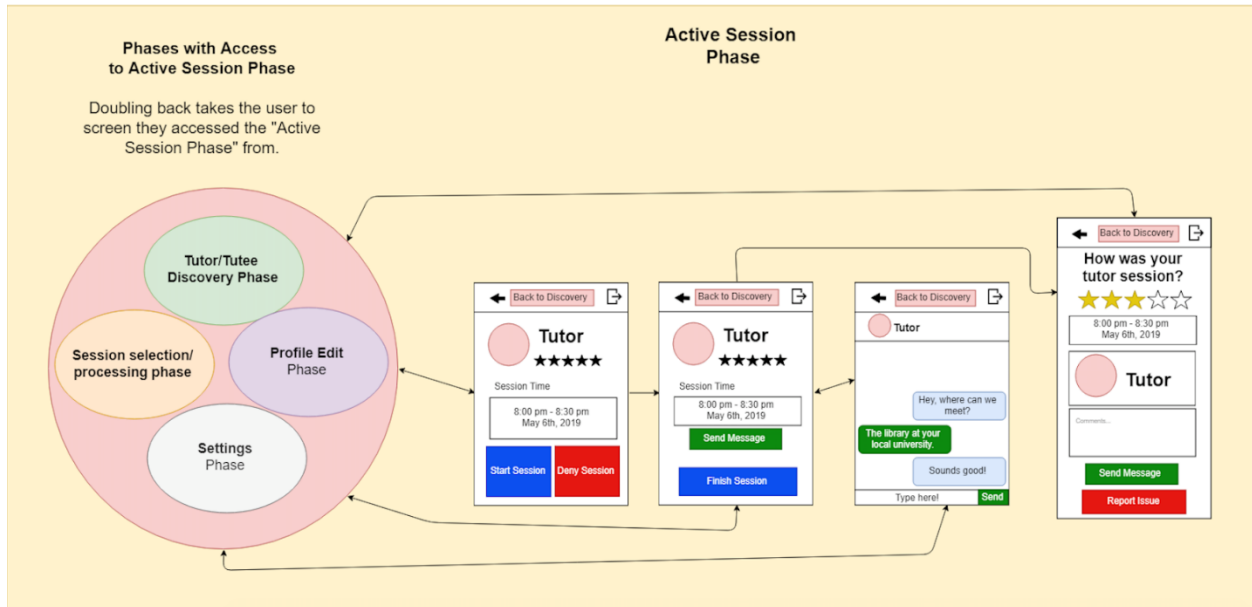


Figure 10: UI/UX Phase 4A

During an active session, the “Active Session” beacon will be displayed, as shown in Figure 11. Users will be able to toggle in and out of their active sessions from the screen they entered from.

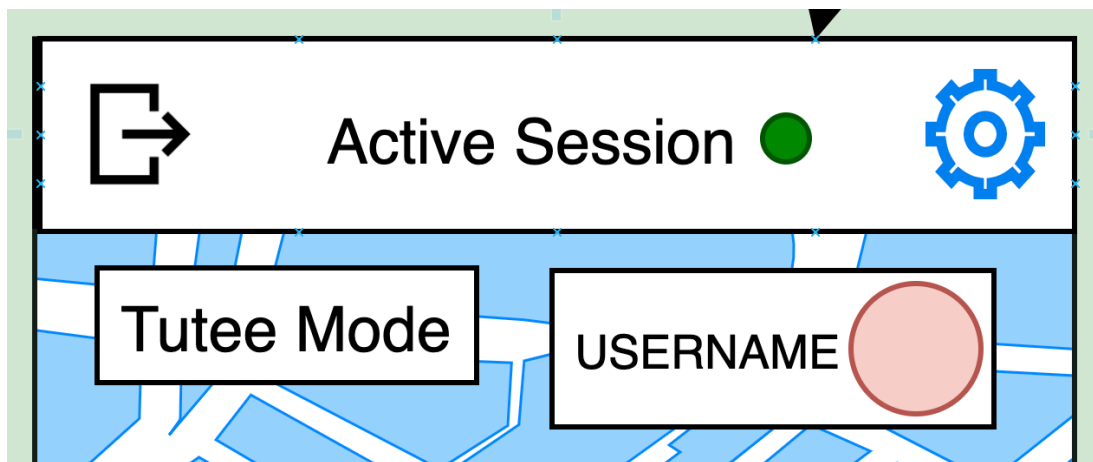


Figure 11: UI/UX Phase 4B

In addition to these four phases, Tutor Dash will feature two views that allow the user to personalize their account. The first of these views, as shown in Figure 12, is the settings view,

where the user will be able to modify their notifications settings, view their calendar, view their profile, review the terms of use policies, view the user manual, become a tutor if they have not already done so, and send out notifications to nearby tutees.

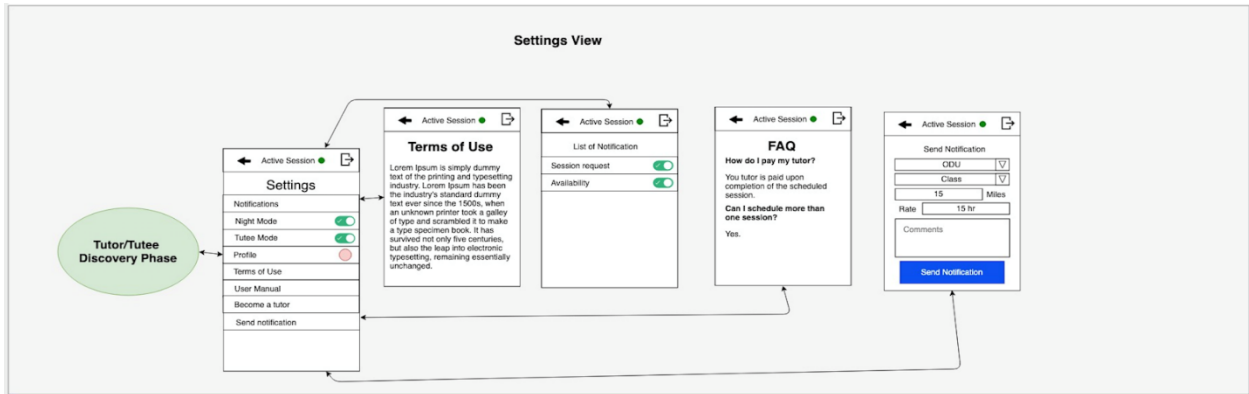


Figure 12: UI/UX View 1 - Settings

The profile edit view, as shown in Figure 13, will allow the user to view and change information regarding their account, such as their username, e-mail, biography, tutored classes, scheduled sessions, session requests, previous sessions, and availability schedule. Additionally, they can view, but not change, their personal ratings. Both views will be accessible from the tutor/tutee discovery phase.

[This space intentionally left blank]

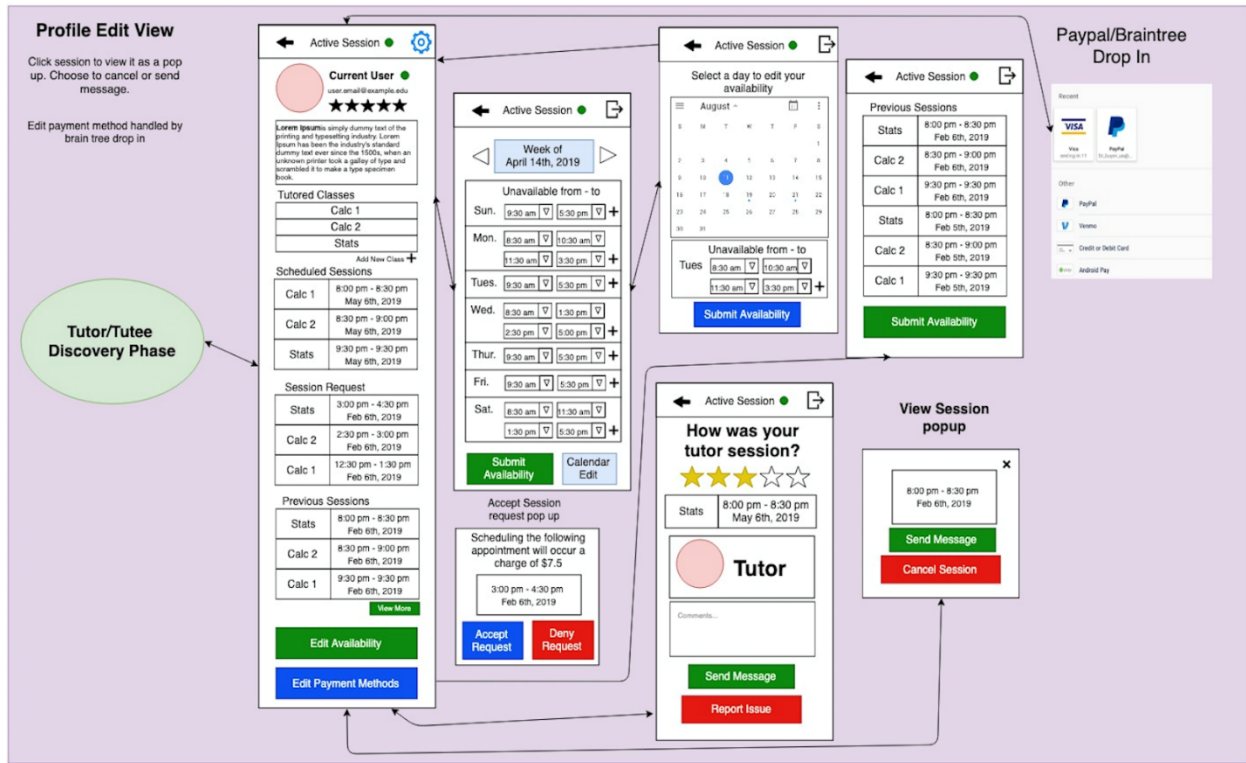


Figure 13: UI/UX View 2 – User Profile

4.2.3 Algorithms

The PDF transcripts parser algorithm is necessary to determine which classes a user is qualified to tutor and to add those courses to the database. This algorithm will require the user to submit a transcript PDF and will require usage of the PDFBox Java library. The logic is shown in Figure 14.

[This space intentionally left blank]

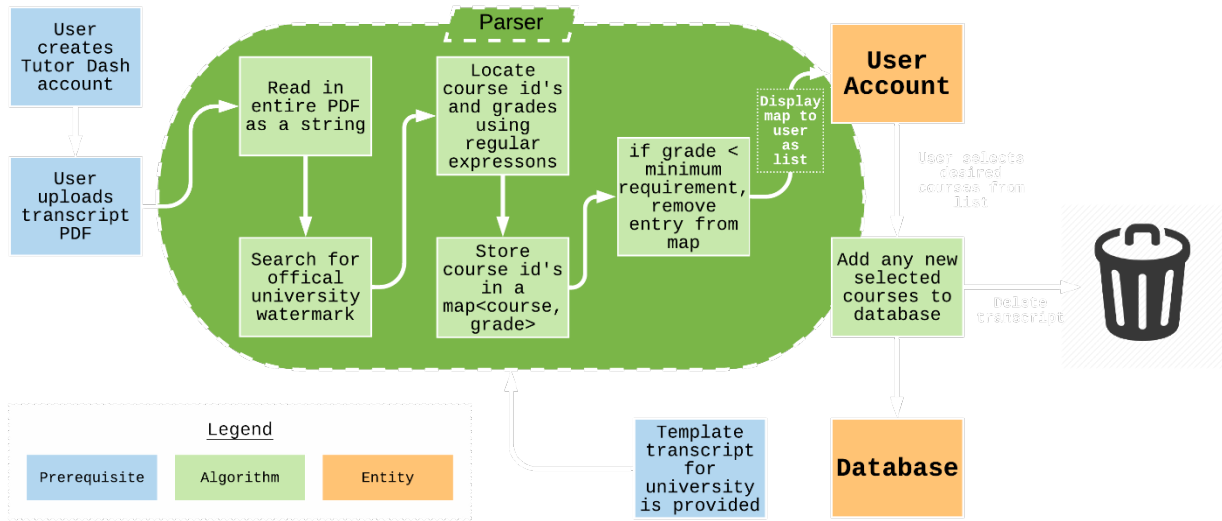


Figure 14: PDF Transcript Parser Algorithm Flow

The distance estimator algorithm will be required to display a set of users within a certain distance of a user that enables the feature. Utilizing the user’s Android device’s GPS coordinates, the distance information will be updated as often as possible. The logic is shown in Figure 15.

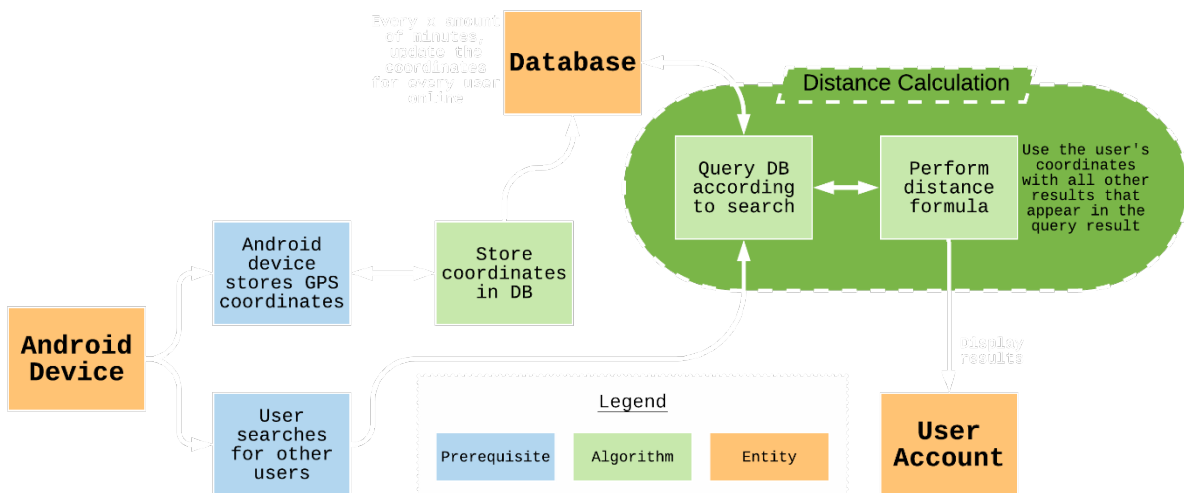


Figure 15: Relative Distance Calculator Algorithm Flow

The pay-rate calculator algorithm will be used to keep rates competitive by establishing an upper bound on the price of a tutoring session. It will be based on several factors, including the tutor rating, both overall and for the specific course, course demand, and time of day. The logic is shown in Figure 16.

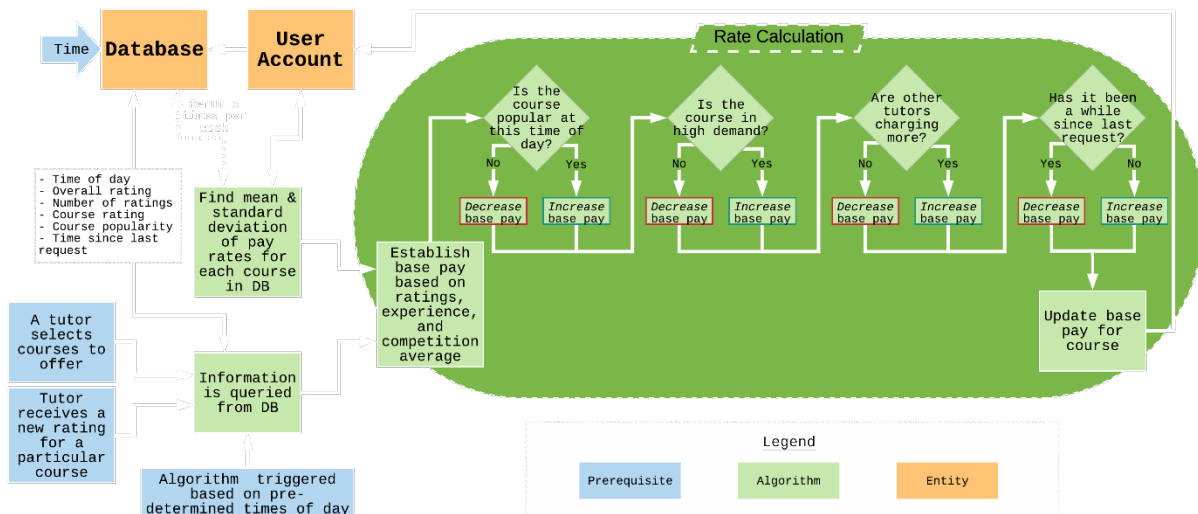


Figure 16: Competitive Pay-Rate Algorithm Flow

A web conference creator algorithm will be used to automatically create a Google Hangouts meeting if a session is chosen to be held via web conferencing. This algorithm will require use of the Selenium Webdriver. The logic is shown in Figure 17.

[This space intentionally left blank]

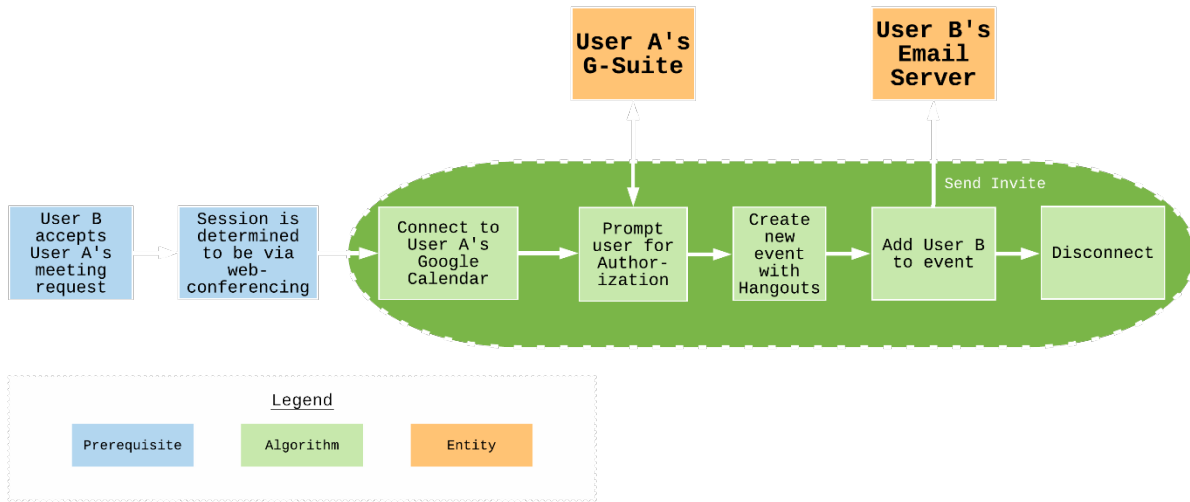


Figure 17: Web Conference Appointment Creator Algorithm Flow

Tutor Dash will have a payment logic algorithm to ensure that in-app payments are handled correctly. The logic is shown in Figure 18.

[This space intentionally left blank]

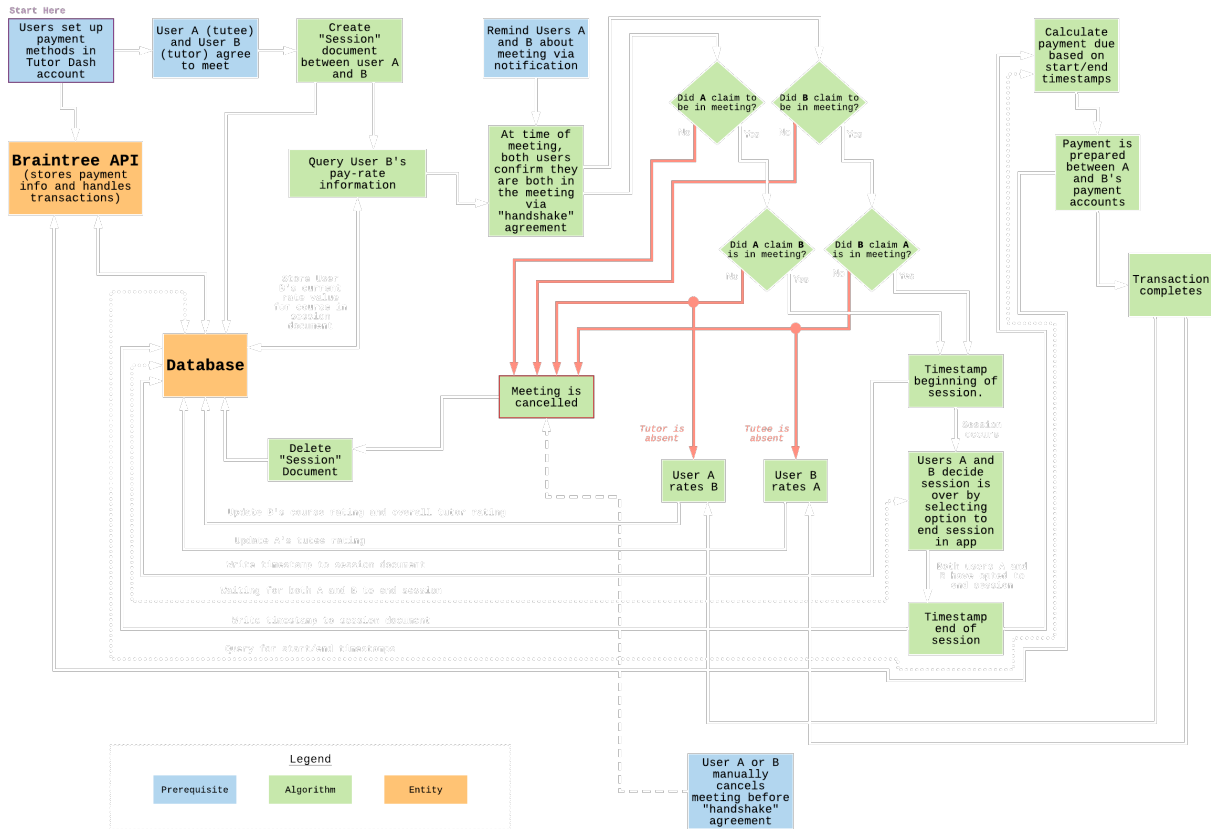


Figure 18: Payment Logic Algorithm Flow

4.3 Prototype Features and Capabilities

The Tutor Dash prototype will be an application that allow student tutors and tutees to connect to their counterparts. Features such as making payment, organizing meetings, and communication between users will be handled by the application as much as possible to increase the ease of use.

The prototype will inherently solve the problem presented in section 1 by creating a centralized platform where tutors and tutees can quickly find each other and set up a tutoring session. The prototype will differ from the real-world product in a few ways, as shown in Table 1.

4.4 Risks and Mitigations

Figures 19 and 20 make up Tutor Dash’s risk matrix. Tutor Dash’s PDF transcript parser will prevent potential tutors from faking their qualifications. The application’s rating system will discourage poor tutors and tutees and will help prevent abuse. Furthermore, to prevent users from leaving false negative reviews, Tutor Dash will allow all users to challenge reviews they believe are not truthful. The Google Calendar API will not allow users to book overlapping sessions.


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

Figure 19: Risk Matrix Overview

ID	Risk Description	Mitigation(s)
C1	Student finds tutors to be unhelpful	<ul style="list-style-type: none"> • Rating system • Payment refunds
C2	Prospective tutors faking their qualifications	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Give small bonuses to tutors for a limited time (similar to Uber’s business model)
C4	Shortage of tutees	<ul style="list-style-type: none"> • Give free sessions to new users • Give loyaltee-free sessions for a certain number of usages
C5	Tutor/tutee leaves a false negative review	<ul style="list-style-type: none"> • Allow users to challenge reviews (requires manual investigation) • Withhold ratings/reviews until both users agree on justification
C6	Users abuse application; use app maliciously	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Preallocate payments • Require deposits • Threat of poor ratings
C9	Users try to book overlapping sessions	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Alert users of the minimum requirements for web conference meetings upon selecting ‘web conferencing’ as a tutoring preference.
T1	Payment is not received	<ul style="list-style-type: none"> • Integrate usage of a 3rd party API designed to handle e-transactions • Braintree
T2	Difficulty automating the process of reading a submitted transcript	<ul style="list-style-type: none"> • Define reusable code for general case • Optimize as more information is discovered
T3	Database server failure	<ul style="list-style-type: none"> • Use reliable servers maintained by large corporations • Firebase
T4	Security breach	<ul style="list-style-type: none"> • Use 3rd party APIs which are already secure
T5	Application is not compatible on all android devices	<ul style="list-style-type: none"> • Define minimum SDK for weaker hardware phones • Define normal SDK for normal hardware phones
T6	Network server failure	<ul style="list-style-type: none"> • Server redundancy
T7	Pay-rate algorithm doesn’t calculate competitive rates	<ul style="list-style-type: none"> • Determine a base pay that will increase/decrease due to various factors

		<ul style="list-style-type: none"> • Compare pay-rates of similarly rated tutors who tutor the same courses
T8	Web-conferencing session is not set up properly	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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)	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Terms of use agreement • Reporting features

Figure 20: Risk Matrix Descriptions

[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 incompleteness 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 focus solely in that space.

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

Tutee: A university student seeking academic assistance.

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

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

6 References

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

URL: <https://www.hanoverresearch.com/wp-content/uploads/2017/08/Academic-Tutoring-in-Comprehensive-Universities.pdf>

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

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

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

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

URL: www.odu.edu/success/academic/tutoring/course-specific

"Courses of Instruction." Old Dominion University, Feb. 2019.

URL: catalog.odu.edu/courses/

Ciscell, Galen, et al. "Barriers to Accessing Tutoring Services Among Students Who Received a MidSemester Warning." ERIC, Pacific Lutheran University - Department of Sociology, 2016. URL: files.eric.ed.gov/fulltext/EJ1114513.pdf

Evans MDR, Kelley P and Kelley J (2017). Identifying the Best Times for Cognitive Functioning Using New Methods: Matching University Times to Undergraduate Chronotypes. *Front. Hum. Neurosci.* 11:188. doi: 10.3389/fnhum.2017.00188.
URL: https://www.frontiersin.org/articles/10.3389/fnhum.2017.00188/full?utm_source=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

"Peer Assisted Learning" BMC Education, 8 March 2006

URL: <https://bmcmmededuc.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=JnNpdGU9ZWwhvc3QtbGl2ZSZzY29wZT1zaXRI#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://bmcmmededuc.biomedcentral.com/articles/10.1186/1472-6920-14-115>
<https://bmcmmededuc.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=true>

"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