Tutor Dash

“Find a tutor. Be a tutor.”

Prototype Design

Old Dominion University
CS410 Spring 2019
Team Gold
May 8, 2019
# Table of Contents

1. **Feasibility** ........................................................................................................... 3
   - Team .................................................................................................................. 3
   - Background ....................................................................................................... 4
   - The Problems ................................................................................................... 7
   - Current Process Flows ..................................................................................... 8
   - Our Solution ..................................................................................................... 10
   - Solution Process Flows ................................................................................. 12
   - What Tutor Dash Won’t Do ............................................................................. 14
   - Competition .................................................................................................... 15
   - Customers & End Users ............................................................................... 16
   - User Roles & Stories ...................................................................................... 18

2. **Design** .................................................................................................................. 22
   - Major Functional Components ................................................................. 22
   - Development Tools ....................................................................................... 23
   - Work Breakdown Structure ............................................................................ 24
   - Core Components ........................................................................................... 25
     - UI/UX ......................................................................................................... 26
     - Database ................................................................................................. 42
     - Algorithms ............................................................................................... 48

3. **Prototype** ............................................................................................................ 63
   - RWP vs. Prototype ......................................................................................... 63
   - Prototype MFCD ............................................................................................. 64
   - Testing Approaches ....................................................................................... 65
   - Development Model ...................................................................................... 68
   - Agile Sprints ................................................................................................. 69
   - Development Schedule ............................................................................... 70
   - Benefits ......................................................................................................... 71
   - Long-Term Goals ........................................................................................... 72
   - Conclusion .................................................................................................... 73

4. **References** ........................................................................................................ 74

- Hardware System Reqts. .................................................................................. 57
- Risk Analysis ..................................................................................................... 58
  - Customer Risks ............................................................................................. 59
  - Technical Risks ............................................................................................. 61
  - Legal Risks ..................................................................................................... 62

---

CS410 - Team Gold  
Design Prototype - Spring 2019

Old Dominion University

05/08/2019
Availability of University Tutoring Programs

- University students study at no specific time of the day.
- Studies even suggest that more students study at night (as opposed to daytime).

Distribution of university students' study times

The Private Tutoring Market

- Private tutoring is a growing industry.
- However, there is no centralized network for tutors.
- Most tutor-hosting services only tend to focus in the online tutoring market.

Source: technavio.com
University Students as Tutors

- Student-to-student interactions are effective; tutees learn more cognitively through bonds with their peers.
- PAL is a peer mediated instructional program where students tutor their peers.
- Peer Assisted Learning Programs (PAL) proves that students tutoring other students is effective.
- University students are willing to tutor people what they know in their free time if it is easy to find people to tutor.
The Problems

1. Tutoring services available to university students are limited in scope, do not provide flexibility, and lack a centralized platform for promotion.

2. Students willing to provide tutoring services lack a tool to promote those services.

Problem Characteristics

- Limited Scope; not all courses offer tutoring
- Tutoring services do not encompass all study times.
- Information inconsistency; university tutoring information is spread out
- Students can’t always trust private tutors to understand their course-specific material
- Students do not have an adequate platform to advertise their tutoring services.
Current Process - Tutees Seeking Tutors

University student seeks tutoring services

Student's university provides adequate tutoring resources

Course specific tutoring isn't offered

Tutoring availability is limited

Tutoring information is contradictory

Student seeks tutoring elsewhere

Many options

Online tutor-hosting platform

Social Media

Flyers

Student finds tutor

Do the student's and tutor's availability align?

Yes

Is the tutor local to the student?

Yes

Does the tutor understand student's specific course?

Yes

Tutoring session is scheduled

Waits

At scheduled appointment, did student and tutor meet?

Yes

Student pays and rates tutor

No

Session occurs

No

Yes

Does the tutor offer online options?

Yes

No

Student does not find tutor

Reasons why student may not find adequate tutoring services provided by their university

Legend

Starting/Ending points
University tutoring deficiencies
Undesirable events
Desirable events
Solvable issues
General events

Student understands course material
Current Process - Tutors Seeking Tutees
Proposed Solution - Tutor Dash

- Tutor Dash is a mobile application-based service that unifies university students who are interested in tutoring with university students who desire course-specific tutoring in real-time.

- Tutor Dash provides an extension of any university's current tutoring services so that students can receive help for all courses 24/7.
Characteristics of Tutor Dash

- User-base constrained to university students
- Every tutor verified based on their previously taken classes
- Rating system for both tutors and tutees
- Qualified tutors can tutor in any class available at their respective university
- Tutors and tutees can be available at any time
- Notifications sent to connect participating tutors and tutees
- Allows for in-person and online meetings
- Mechanism for payment within application
How Tutor Dash Affects Current Processes

- Creates a more obvious connection between tutors and tutees at universities
- Niche tool for participating tutors and tutees
- More centralization → Less options → More opportunity
Current Process Flows & Tutor Dash

Initial Setup (Before usage)

University student wants to offer private tutoring services

Download Tutor Dash on smartphone

Input basic user information

Request academic transcript from university registrar

Upload transcript

Tutor Dash

Tutors get notified when potential tutees are available in the area.

Send out notifications to potential tutees

Tutoring session is scheduled in real-time

Minimal Waiting

Tutoring session occurs

Tutor receives payment

Tutee rates tutor

Tutees are notified when potential tutors are available in the area.

Send out notifications to local tutors
What Tutor Dash Will Not Do

- Not an “official” university tutoring resource, rather a tool for presenting information on a unique platform
- Will not make any money off of existing university tutoring resources
- Will not interfere with current university tutoring resources; only improve advertisement to extend reach
- Will not violate the Family Education Rights and Privacy Act (FERPA)
- Cannot guarantee there is a tutor for everyone online 24/7
- Cannot prevent students from voluntarily violating the honor code
- Will not be a hosting platform for established tutoring businesses
## Competition

<table>
<thead>
<tr>
<th>Features</th>
<th>US</th>
<th>DIRECT COMPETITORS</th>
<th>INDIRECT COMPETITORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offers various university course-specific tutoring</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Allows qualified undergraduate students to be tutors</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Sends notifications about local tutors/tutees</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Qualified tutors are constrained to university communities</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Provides real-time scheduling capabilities</td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td>Available as mobile application</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Includes online tutoring options</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Includes in-person tutoring options</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Provides tutor ratings</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Establishes hourly rate ceilings</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Requires tutor verification/validation</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Does not require subscription commitment</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Provides 24/7 scheduling</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
Customers

- University students seeking tutoring services (academic help)
- University students interested in tutoring (providing academic help)
End Users

University student bodies

- University students looking for tutors
  - Seeking immediate help
  - Seeking future help

- University students looking for tutees
  - Seeking network growth
  - Seeking extra money
## User Roles

<table>
<thead>
<tr>
<th>Tutee</th>
<th>Private Tutor</th>
<th>Tester</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Tutee" /></td>
<td><img src="image" alt="Private Tutor" /></td>
<td><img src="image" alt="Tester" /></td>
</tr>
</tbody>
</table>
| - University student  
- Seeks academic assistance | - University student  
- Qualified to tutor previously taken courses  
- Offers tutoring services  
- Works independently | - Team member  
- Responsible for quality of software  
- Uses tools to write tests for test automation |

*CS410 - Team Gold  
Design Prototype - Spring 2019*
User Stories - Tutee

As a tutee, I need...

1. ...the opportunity to receive tutoring in any class which I am enrolled in.
2. ...the ability to receive/request tutoring at any time of the day.
3. ...the ability to message tutors in real-time before hiring them.
4. ...to search for classes I need help in and receive a list of tutors.
5. ...payments to be secure and only charged when I can confirm that the meeting did, in fact, occur.
6. ...the ability to leave meaningful ratings and reviews based on my experiences with tutors.
7. ...the ability to report malicious activity regarding authorized tutors' activities.
8. ...the ability to send out alerts so that tutors will be notified when I am looking for them.
9. ...the ability to confirm that a session did, in fact, occur.
10. ...to see some indication that tutors are currently available to hire right now.
11. ...to see the distance a tutor is away from me if I am, in fact, seeking an in-person meeting.
12. ...the ability to receive tutoring in-person.
13. ...the ability to receive tutoring online.
14. ...the ability to search for tutors tutoring any course at my university.
15. ...the ability to reconnect with a tutor I liked so I can rehire them in the future.
16. ...tutor profiles to be public when I search for them.
17. ...sensitive account information to be private and secure.
18. ...reassurance that the person I am meeting is the person they say they are.
19. ...registered tutors to be qualified and authorized to tutor me.
20. ...to search for tutors who attend my university within a certain radius.
21. ...the ability to report malicious activity regarding authorized tutors' activities.

As a tutee I wish...

1. ...for an appealing visual display both for daytime AND nighttime hours.
2. ...for my payment methods to be remembered.
3. ...that if I am searching for a course with no tutors in the system, I will be alerted.
4. ...for a way of informing Tutor Dash if my desired course is not in the system, so they can alert potential tutors.
5. ...for a refund on my online session if there are technical difficulties preventing my session from occurring properly.
User Stories - Private Tutor

As a private tutor, I need...
1. ...my qualifications to be based on my previous coursework.
2. ...to receive notifications when potential tutees message/hire me.
3. ...my pay-rate to be as competitive as possible so I don’t wait too long to get hired.
4. ...the option to receive alerts when potential tutees in the network are seeking tutoring in courses I tutor.
5. ...a calendar to maintain and update my availability at any time, which potential tutees can publicly view.
6. ...the ability for potential tutees to view my user profile.
7. ...the ability for potential tutees to message me before scheduling a session.
8. ...my ratings to be given only by tutees which I have tutored.
9. ...a mechanism for confirming a session did, in fact, occur.
10. ...payment to be handled within the application.
11. ...my charging pay-rate to be based partly on my experience.
12. ...the ability to challenge poor/negative ratings and/or reviews.
13. ...a mechanism for confirming a session did, in fact, occur.
14. ...the ability to rate tutees based on my experiences with them.
15. ...the ability to refuse service.
16. ...a means of toggling my availability in real-time.
17. ...web-conferencing and in-person tutoring options.
18. ...sensitive account information to be private and secure.

As a private tutor, I wish...
1. ...for a log to keep track of my session and payment history.
2. ...the option to turn off location services if I am only offering online sessions.
3. ...for alerts that tell me classes I don’t tutor, but am eligible to tutor are in high demand.
4. ...an alert every X months to remind me to upload an updated transcript so I can tutor more courses.
5. ...for a bonus to tutor courses in high demand if there is a low supply of tutors.
6. ...for an alert before a scheduled session occurs if payment for that session fails to go through.
User Stories - Tester

As a tester, I need...
1. ...to create a variety of mocked up user accounts with semi-automated decision capabilities to simulate an interactive experience.
2. ...a visual log that shows me all the attributes and results of my interactive simulation based on events that I initiate.
3. ...to simulate the signing up both a tutor AND a tutee.
4. ...to simulate a tutee searching for a tutor.
5. ...to simulate a tutor searching for a tutee.
6. ...to mock up data/accounts to simulate various tutors in various courses that exist at ODU.
7. ...to mock up data/accounts to simulate various tutees who would be using the app.
8. ...to design a series of test cases where a tutor/tutee is rated at various different times.
9. ...to design a plan to ensure that pay-rates are affected by weighted ratings, course demand, and time.
10. ...to simulate a tutee requesting to hire a tutor.
11. ...to simulate a tutor denying AND a tutor accepting a tutee’s request.
12. ...to simulate a tutee who has sent an alert into the network looking for a tutor.
13. ...to simulate a tutee who cancels after hiring a tutor.
14. ...to simulate the transactions that take place before and after an appointment (deposit and payment respectively).
15. ...to supply a series of mocked up transcripts to the transcript parser that register as official to observe tutoring eligibility based on academic history.
16. ...to supply fake transcripts to ensure the security of the transcript parser.
17. ...to simulate a tutor AND tutee query with both inactive and active users existing in the network.
18. ...to simulate a tutee requesting both an in-person and online meeting.
19. ...to simulate what happens with payments in the case of when both a tutor and tutee agree to a web conference, but there are technical difficulties.
20. ...to simulate a user authenticating themselves via email.
21. ...to simulate the re-authentication process triggered by when a user navigates back into the application from an outside view.
22. ...to simulate a conversation among two users at two distinctive points in time to make sure chat history is retrieved.
23. ...to simulate a series of exact queries in which users who are active and appear in the result are moving away/towards the user searching for them.
24. ...to simulate a meeting in which either (but not both) the tutee or tutor never acknowledge the start of the meeting.
25. ...to simulate the event in which a tutee gets refunded based on a poor experience.
26. ...to simulate a scheduling conflict due to the overlap of calendar events.

As a tester, I wish...
1. ...to provide a test case for when users are currently using the application, and the database fails.
2. ...to run my test suites for every unit of work alongside every build.
3. ...for a tool to aid in the automation of database querying.
Major Functional Components
## Development Tools

<table>
<thead>
<tr>
<th>Component</th>
<th>Tool (Android)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Java</td>
</tr>
<tr>
<td>IDE</td>
<td>Android Studio</td>
</tr>
<tr>
<td>UI/UX</td>
<td>Android Studio</td>
</tr>
<tr>
<td>Database</td>
<td>Firebase</td>
</tr>
<tr>
<td>Payment Mechanism</td>
<td>Braintree</td>
</tr>
<tr>
<td>Build Manager</td>
<td>Gradle</td>
</tr>
<tr>
<td>Version Control</td>
<td>GitLab</td>
</tr>
<tr>
<td>Continuous Integration/Deployment</td>
<td>GitLab</td>
</tr>
<tr>
<td>Issue Tracker</td>
<td>Trello</td>
</tr>
<tr>
<td>Testing Framework(s)</td>
<td>JUnit/Espresso/Firebase</td>
</tr>
</tbody>
</table>
Work Breakdown Structure

1. Development Requirements
2. Design
3. Implementation
4. Testing

* Testing and Implementation will be done in parallel
Core Components

UI/UX

Database

Algorithms
UI/UX Breakdown

Tutor Dash’s UI/UX design is broken down into 4 phases and 2 views:

- **Phases**
  1. Login/Sign up
  2. Tutor/Tutee Discovery
  3. Session Selection
  4. Active Session

- **Views**
  1. Settings
  2. Profile Edit

“Phase” implies sequential ordering, while “View” implies no such ordering.
UI/UX Big Picture

- Blue: Login/Sign Up
- Green: Tutor/Tutee Discovery
- Orange: Session Selection
- Yellow: Active Session
- Grey: Settings
- Purple: Profile Edit

CLICK HERE to view enlarged image
Phase 1 - Login/Sign Up

The **Login/Sign-Up Phase** allows the user to do the following:

- Sign up and register as a user (either tutee or tutee and tutor)
- Login to registered account
- Upload transcript for tutor authentication
- Initialize personal list of tutored classes from list of all permitted classes
Phase 1 - Login/Sign Up UI
Phase 2 - Tutor/Tutee Discovery

The Tutor/Tutee Discovery Phase allows users to do the following:

- Search for tutors or tutees based on mode through map or text
- Change view modes from map/text
- Change or sort content of map or text view through search criteria
- Search for a specific tutor in search view
- Change status to or from active
Phase 2 - Tutor/Tutee Discovery UI
Phase 3 - Session Selection

The **Session Selection Phase** gives the user the ability to do the following:

- **View user information:**
  - Username
  - Ratings
  - Biography
  - Classes available for tutoring

- **Send messages to other users**

- **Schedule sessions from a tutor’s available times**

- **Schedule sessions ASAP (in real-time)**
Phase 3 - Session Selection UI

Session selection phase

Tutor/Tutee Discovery Phase

Tutor
15 Sessions

Select a day, time, and method
Year
Month
Day

Scheduling the following appointment will result in a charge of $7.50
6:30 am - 9:30 am Feb 5th, 2019
Accept Charge
Deny Charge

Session Secured!
Talk to your tutor about this session!
Phase 4 - Active Session (Part A)

- The Active Session Phase is accessible from any screen where the “Active Session” beacon is presented.
- The beacon’s availability is dependent on the user’s scheduled sessions.
- Users may toggle in and out of their active sessions to and from the phase/screen they entered from.
Phase 4 - Active Session (Part B)

The Active Session Phase gives the user the ability to do as follows:

- Begin scheduled tutoring session.
- Communicate through in-app messaging system to coordinate session details with tutor.
- Dictate end of tutoring session upon completion.
- Rate tutor out of 5 stars and review them for the recently-finished session.
- Report issue with current session and/or tutor.
Phase 4 - Active Session UI

Phases with Access to Active Session Phase

- Tutor/Tutee Discovery Phase
- Session selection/processing phase
- Profile Edit Phase
- Settings Phase

Active Session Phase

- Tutor
  - Session Time: 8:00 pm - 8:30 pm, May 6th, 2019
  - Start Session, Finish Session
- Tutor
  - Review session
  - Send Message
  - Sounds good!
- Tutor
  - Library at your local university
  - Sounds good!
  - Send

How was your tutor session?

5 stars:
- Tutor
- 8:00 pm - 8:30 pm
- May 6th, 2019

Double back takes the user to screen they accessed the "Active Session Phase" from.
View 1 - Options & Settings

The Options and Settings View gives the user the ability to do as follows:

- Modify Notifications
- Toggle Night Mode
- View Calendar for future sessions
- View their profile
- View the ‘Terms of Use’ policies
- View user’s manual
- Become a tutor
- Send out local notifications
View 2 - Profile Edit

The Profile Edit View gives the user the ability to do as follows:

- View their account information:
  - Username
  - Email
  - Rating
  - Biography
  - Tutored classes
  - Scheduled session(s)
  - Session request(s)
  - Previous session(s)
  - Availability

- Modify any of the aforementioned elements other than personal rating
View 2 - Profile Edit UI
Database
Firebase

- **Firebase** is a mobile and web application development platform.
- Tightly integrated with Google Cloud Platform
- Tutor Dash will use the following Firebase products to handle backend services:
  - *Cloud Firestore* - Database
  - *Firebase Auth* - User Authentication
  - *Cloud Storage* - File Storage

Used By:

[Image of logos used by various companies]
Cloud Firestore

- NoSQL document-oriented database
- Keeps data synced across all client devices in real time
- Designed for automatic scalability
- Database will store user information, school information, schedules, messaging, and reviews.
• Blacklisted users’ emails are added to the ‘Blacklist’ table once they get banned. Their account will subsequently be deleted.

• Payments options are not stored in the DB, but payment history is stored.
User Authentication & Cloud Storage

● **User Authentication**
  ○ Authentication allows users to create an account using email and password.
  ○ Capable of sending email confirmation to activate account.
  ○ Integrates with Cloud Firestore for secure DB access.

● **Cloud Storage**
  ○ Capable of storing user generated content such as photos and videos.
  ○ High scalability. Same infrastructure that powers Spotify and Google Photos.
Braintree - Payments

- Collection of cross-platform tool sets that allow developers to include multiple payment options safely within their applications
- Include drop-in UIs
- Managed by PayPal
- Used by several well-known companies such as Uber and Poshmark
- **Required toolsets**
  - Braintree Java SDK
  - Braintree Android v3 SDK
Algorithms
PDF Transcript Parser

- **Purpose**
  - Determine which classes a user is qualified to tutor
  - Add new courses being tutored to the database

- **Tools**
  - PDFBox Java Library

- **Parameters**
  - University name
  - Transcript (PDF)
  - Minimum qualifying grade
PDF Transcript Parser - Logic Flow

User creates Tutor Dash account

User uploads transcript PDF

Parser

Read in entire PDF as a string

Locate course id's and grades using regular expressions

if grade < minimum requirement, remove entry from map

Search for official university watermark

Store course id's in a map<course, grade>

Display map to user as list

User Account

User selects desired courses from list

Add any new selected courses to database

Delete transcript

Database

Template transcript for university is provided

Legend

Prerequisite

Algorithm

Entity
Pay-Rate Calculator

- **Purpose**
  - Keep pay-rates competitive by providing an upper bound based on various factors
  - Alleviate the possibility of tutors not getting hired often enough

- **Parameters**
  - Tutor rating (course-specific)
  - Tutor rating (overall)
  - Course demand/popularity
  - Tutor’s Experience
  - Time of day
  - Mean & standard deviation of rates for courses
  - Time since tutor’s last request in that course
Pay-Rate Calculator - Logic Flow

1. **User Account**
   - Time of day
   - Overall rating
   - Number of ratings
   - Course rating
   - Course popularity
   - Time since last request
   - Find mean & standard deviation of pay rates for each course in DB
   - Return x times per week

2. **Rate Calculation**
   - Is the course popular at this time of day?
     - No → Decrease base pay
     - Yes → Increase base pay
   - Is the course in high demand?
     - No → Decrease base pay
     - Yes → Increase base pay
   - Are other tutors charging more?
     - No → Decrease base pay
     - Yes → Increase base pay
   - Has it been a while since last request?
     - Yes → Increase base pay
     - No → Decrease base pay

3. **Database**
   - A tutor selects courses to offer
   - Tutor receives a new rating for a particular course
   - Information is queried from DB
   - Algorithm triggered based on pre-determined times of day

4. **Legend**
   - Prerequisite
   - Algorithm
   - Entity
Relative Distance Estimator

- **Purpose**
  - Display how far a set of users B is from user A based on user A’s search results.
  - Keep the distance information updated as often as possible.

- **Parameters**
  - Time
  - Time interval for updates
  - Android device’s GPS coordinates
    - User A’s longitude/latitude
    - Every user in B’s longitude/latitude
Relative Distance Estimator - Logic Flow

Every x amount of minutes, update the coordinates for every user online.

1. **Database**
   - Query DB according to search
   - Perform distance formula
   - Use the user's coordinates with all other results that appear in the query result
   - Display results

2. **Distance Calculation**

3. **User Account**

4. **Legend**
   - Prerequisite
   - Algorithm
   - Entity

5. **Android Device**
   - Android device stores GPS coordinates

6. **User**
   - User searches for other users

---

Old Dominion University
05/08/2019
Web Conference Creator

- **Purpose**
  - Create a google hangouts meeting for two user’s if the tutoring session in question is via web-conferencing

- **Tools**
  - Selenium Webdriver (Java)

- **Parameters**
  - Start time of session
  - End time of session
  - User email addresses
    - Tutor
    - Tutee
Web Conference Creator - Logic Flow

- Tutor accepts tutee's request
- Session is determined to be via web-conferencing
- Open web-browser with hidden window on server
- Navigate to Google Calendar
- Create new event
- Invite tutee and tutor to event

Legend:
- Prerequisite
- Algorithm
- Entity

Tutee's Email Address
Send Invite

Tutor's Email Address
Send Invite
Hardware System Requirements

- Android API 16 (JELLY_BEAN) is required for both Firebase and Braintree
- Devices must have GPS capabilities
Risk Matrix

- “C” → Customer Risks
- “T” → Technical Risks
- “L” → Legal Risks

<table>
<thead>
<tr>
<th>Probability</th>
<th>Very Low</th>
<th>Low</th>
<th>Moderate</th>
<th>High</th>
<th>Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very High</td>
<td>T3, T4, L1</td>
<td>T6, L2</td>
<td></td>
<td></td>
<td>C3, C4</td>
</tr>
<tr>
<td>High</td>
<td>T1, C7</td>
<td>C6, T9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moderate</td>
<td>C5</td>
<td>T8</td>
<td></td>
<td></td>
<td>C1, C8, T7</td>
</tr>
<tr>
<td>Low</td>
<td>C2, T5</td>
<td></td>
<td></td>
<td></td>
<td>C10, T2</td>
</tr>
<tr>
<td>Very Low</td>
<td></td>
<td>C9</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Risks - Customer Risks (Part 1)

<table>
<thead>
<tr>
<th>ID</th>
<th>Risk</th>
<th>Mitigation(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Student finds tutors to be unhelpful</td>
<td>● Rating system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Payment refunds</td>
</tr>
<tr>
<td>C2</td>
<td>Prospective tutors faking their qualifications</td>
<td>● Require official transcript from university registrar</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Make tutors only eligible to tutor classes he/she has received a B or higher in</td>
</tr>
<tr>
<td>C3</td>
<td>Shortage of tutors</td>
<td>● Give small bonuses to tutors for a limited time (similar to Uber’s business model)</td>
</tr>
<tr>
<td>C4</td>
<td>Shortage of tutees</td>
<td>● Give free sessions to new users</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Give loyalty-free sessions for a certain number of usages</td>
</tr>
<tr>
<td>C5</td>
<td>Tutor/tutee leaves a false negative review</td>
<td>● Allow users to challenge reviews (requires manual investigation)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Withhold ratings/reviews until both users agree on justification</td>
</tr>
<tr>
<td>ID</td>
<td>Risk</td>
<td>Mitigation(s)</td>
</tr>
<tr>
<td>----</td>
<td>----------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| C6 | Users abuse application; use app maliciously                         | • Require users to agree to the terms of use agreement  
• Blacklist (ban) users who violate 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. |
## Risks - Technical Risks

<table>
<thead>
<tr>
<th>ID</th>
<th>Risk</th>
<th>Mitigation(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Payment is not received</td>
<td>● Integrate usage of a 3rd party API designed to handle e-transactions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Use Braintree</td>
</tr>
<tr>
<td>T2</td>
<td>Difficulty automating the process of reading a submitted transcript</td>
<td>● Define reusable code for general case</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Optimize as more information is discovered</td>
</tr>
<tr>
<td>T3</td>
<td>Database server failure</td>
<td>● Use reliable servers maintained by large corporations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Use Firebase</td>
</tr>
<tr>
<td>T4</td>
<td>Security breach</td>
<td>● Use 3rd party APIs which are already secure</td>
</tr>
<tr>
<td>T5</td>
<td>Application is not compatible on all android devices</td>
<td>● Define minimum SDK for weaker hardware phones</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Define normal SDK for normal hardware phones</td>
</tr>
<tr>
<td>T6</td>
<td>Network server failure</td>
<td>● Server redundancy</td>
</tr>
<tr>
<td>T7</td>
<td>Pay-rate algorithm doesn't calculate competitive rates</td>
<td>● Determine a base pay that will increase/decrease due to various factors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Compare pay-rates of similarly rated tutors who tutor the same courses</td>
</tr>
<tr>
<td>T8</td>
<td>Web-conferencing session is not set up properly</td>
<td>● Use Google Hangouts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Use one or more G Suite hosts operated by Tutor Dash</td>
</tr>
<tr>
<td>T9</td>
<td>Unexpected interruption prohibits online sessions from occurring</td>
<td>● Refund payments in this case as long as both parties arrived to the meeting.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● Use a Google Hangouts, a commonly used web conference tool maintained by a large corporation.</td>
</tr>
</tbody>
</table>
# Risks - Legal Risks

<table>
<thead>
<tr>
<th>ID</th>
<th>Risk</th>
<th>Mitigation(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Violating The Family Education Rights and Privacy Act (FERPA)</td>
<td>- No portal access&lt;br&gt;- Transcripts are analyzed then thrown out&lt;br&gt;- Users agree to grade disclosure in terms of use agreement</td>
</tr>
<tr>
<td>L2</td>
<td>Users use application for illegal activities</td>
<td>- Terms of use agreement&lt;br&gt;- Reporting features</td>
</tr>
</tbody>
</table>
# Real World Product vs. Prototype

<table>
<thead>
<tr>
<th>Functional Element</th>
<th>RWP</th>
<th>Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-the-fly tutor qualification based on transcript</td>
<td>Fully-Functional</td>
<td>Fully-Functional</td>
</tr>
<tr>
<td>University student verification based on email</td>
<td>Fully-Functional</td>
<td>Fully-Functional</td>
</tr>
<tr>
<td>Search results tailored based on tutor/tutee mode</td>
<td>Fully-Functional</td>
<td>Fully-Functional</td>
</tr>
<tr>
<td>Real-time scheduling</td>
<td>Fully-Functional</td>
<td>Fully-Functional</td>
</tr>
<tr>
<td>Weighted ratings for every course</td>
<td>Fully-Functional</td>
<td>Fully-Functional</td>
</tr>
<tr>
<td>Reviews and comments on user profiles</td>
<td>Fully-Functional</td>
<td>Fully-Functional</td>
</tr>
<tr>
<td>In-app payments/deposits (any transactions)</td>
<td>Fully-Functional</td>
<td>Fully-Functional</td>
</tr>
<tr>
<td>In-app messaging/history of conversations</td>
<td>Fully-Functional</td>
<td>Fully-Functional</td>
</tr>
<tr>
<td>Web conference and in-person meeting support</td>
<td>Fully-Functional</td>
<td>Fully-Functional</td>
</tr>
<tr>
<td>Relative distance user A is from user B appears in query</td>
<td>Fully-Functional</td>
<td>Fully-Functional</td>
</tr>
<tr>
<td>Night mode</td>
<td>Fully-Functional</td>
<td>Fully-Functional (- Necessary since Tutor Dash targets both day and night crowds)</td>
</tr>
<tr>
<td>Automated pay rate calculation for every course</td>
<td>Fully-Functional</td>
<td>Partially Functional (- Mean &amp; std. dev. of pay-rates will need to be mocked up)</td>
</tr>
<tr>
<td>Reporting features</td>
<td>Fully-Functional</td>
<td>Partially Functional (- Users can report, but no action will occur)</td>
</tr>
<tr>
<td>Re-authentication when navigating back into app</td>
<td>Fully-Functional</td>
<td>Fully-Functional</td>
</tr>
<tr>
<td>Refunds due to poor experiences</td>
<td>Fully-Functional</td>
<td>Partially Functional (- However, this feature may disrupt the user experience)</td>
</tr>
<tr>
<td>Free sessions/monetary bonuses</td>
<td>Fully-Functional</td>
<td>Eliminated</td>
</tr>
<tr>
<td>Blacklisting of users</td>
<td>Fully-Functional</td>
<td>Eliminated (- Unnecessary for prototype)</td>
</tr>
<tr>
<td>Support of multiple universities</td>
<td>Fully-Functional</td>
<td>Eliminated</td>
</tr>
<tr>
<td>Cross-platform support</td>
<td>Eliminated</td>
<td>Eliminated</td>
</tr>
<tr>
<td>Firebase console linked to test suite(s) with mockups</td>
<td>Eliminated</td>
<td>Fully-Functional</td>
</tr>
</tbody>
</table>
Prototype MFCD

- All major functional components remain
- User-base constrained to ODU students
- Hardware constrained to Android smartphones
Approach: Test each core component independently first.

- **UI/UX**
  - **Espresso** - Built into Android Studio, this serves as the primary tool that Android developers use to automate the testing of UI’s.

- **Database**
  - **Firebase Test Lab** - A cloud-based app-testing infrastructure that allows developers to view results including logs, videos, and screenshots in the Firebase console.
  - **JUnit** - Will be used to query the database for expected output.

- **Algorithms**
  - **JUnit** - A popular testing framework for Java that allows developers to automate tests and integrate them into the build when necessary.
Testing - Integration Testing

**Approach:** Merge core components one by one to ensure compatibility.

- Most likely will be automated using a combination of the unit testing tools.
- Since the major functionality of Tutor Dash is the database, Firebase Test Lab will be used to query information which our team will mockup to test the behaviors of the queries.
- Eventually, once all core components are integrated into the build, these tests evolve into system tests.
Testing - System Testing

**Approach:** Test the system as a whole using virtual & physical devices.

- **Firebase Test Lab**
  - Tutor Dash’s networking functionality is almost entirely dependent on the database. Because of this, the database must be tested extensively to ensure that it will run on all target devices.

- **Android Emulator**
  - Android Emulator is a part of Android Studio that allows developers to choose from a variety of emulators. This will ensure that Tutor Dash fits our minimum SDK requirements for usage.
Development Model - Agile

- Work completed in brief time iterations (sprints)
- Testing implemented throughout development
- Focus on observable behavior

Source: Intelegrain Technologies
Agile Sprints

The Tutor Dash prototype will be divided up into 8 sprints over 11-16 weeks, each with their own focus:

1. **Environment Setup**: Set up DB and make sure we can query information.
2. **UI/UX**: A substantial part of the UI will exist. Incorporate that into backend.
3. **Link Front/Back-Ends**: Payments/User Auth. will be nearing completion. Significant progress on algorithms. Link UI with DB and test.
4. **Algorithms**: Keep making progress on algorithms. Continue testing algorithms and DB.
5. **UI/UX & Testing**: All UI/UX phases/views will exist in one form or another. Keep testing, and incorporate into DB.
6. **Database Finalization**: Finalize the DB server and finish its testing. Begin system tests.
7. **UI/UX Completion**: Link all backend with every UI/UX frame. Focus on integration testing.
8. **Testing Finalization**: Testing needs to be finalized. Unit tests should be passing, and system tests will reflect functional reqts. and hardware specifications.
# Development Schedule

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S</td>
<td></td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S</td>
<td></td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
<td></td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

F = "Start"  C = "Continue"  F = "Finish"  R = "Refine"

<table>
<thead>
<tr>
<th>Alex</th>
<th>Brandon</th>
<th>John</th>
<th>Duncan</th>
<th>Jamauni</th>
<th>Dwight</th>
<th>Not Assigned</th>
</tr>
</thead>
</table>

CS410 - Team Gold

Design Prototype - Spring 2019

Old Dominion University

05/08/2019
Benefits to Customer/User Base

- Expands the scope of any university’s course-specific tutoring program(s).
- Decreases availability conflicts.
- Tutors aren’t waiting around for responses.
- Reduces frequency of cancellations by providing real-time scheduling.
- As tutoring increases, DFWI (Drop/Fail/Withdraw) rates decrease. Students save GPA and money.
- Students won’t have to go out of their way in search of tutoring services since they are contained conveniently in one place.
- Alleviates frustration in regards to contradictory information. Improves information consistency.
Long-Term Goals

- Various university support
- Active community 24/7
- Every class offered at every university in the system has at least one eligible tutor
- Repeat/long-term customers
In Summary

- Our solution aims to engage university students and help them find what they need in an easier and more convenient format than what currently exists.
- Some students need tutors
- Some students want to tutor
- Tutor Dash makes the connection obvious
References - University Tutoring Resources


2. "Campus Tutoring." Old Dominion University, 19 Jan. 2019. URL: www.odu.edu/success/academic/tutoring#tab125=0.


4. "Courses of Instruction." Old Dominion University, Feb. 2019. URL: catalog.odu.edu/courses/

References - Student Behaviors


References - Student Behaviors cont.


13. 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=JnNpdGU9ZWhvc3QtbGRpbj0yMDEwMzgyMDIwMjA0NzEyNzg2MDU2MTI1MDM4MDU3NjIyMzUzNzIwOTIzOCZaZWN0aW9ucy0xMjguMjA3OTQ0MzAzMzEwMDU2Mzg2MjM2NjIyMTUwMzI5NzAyMjE1NjIzMTI3Nzc%3D&btsrc=1&AN=114789419&db=ehh.
References - Competition


15. "Find a Local In-Home Tutor Today." HeyTutor, HeyTutor LLC. URL: heytutor.com/.


