1 Introduction

- Background
    - Availability of tutors
    - Benefit of tutors on university campuses: DFWI Rates and major retention
    - Need for consolidation of tutoring resources on campuses and in current market
    - University students are qualified, yet they won't tutor since scheduling and payment are not convenient
- Problems and their characteristics
    - Current tutoring resources on campus and in market don't suffice
    - Existing resources are separated by departments and not centralized, leading to lack of student knowledge and usage of them
    - University students would tutor if it was easier. No centralized platform currently exists to make that happen
    - Figures
        - 1.1, 1.2
- Summarize how Tutor Dash is going to mitigate the problems and how it will do this.

2 Tutor Dash Product Description

- The Solution (Tutor Dash) and characteristics
    - How does Tutor Dash affect the current process flow?
    - Basic usage flow (how is it better than the current process?)
- Tutor Dash will be a mobile android app. How does it make everything easier?
    - Mention the current process flows and where they are flawed
    - Show the current process flows for Tutor Dash and explain why it improves the current processes (for both tutors and tutees)
    - Introduce the figures for current process and explain their flaws. End it by explaining what Tutor Dash will do to fix this.
    - Figures
        - 1.3, 1.4

2.1 Objectives and Goals of Tutor Dash

- Create a product that fills the gaps that exist in the current private tutoring market.
- Create a centralized tutoring platform amongst university students (several university student bodies in the RWP, but only ODU in prototype)
- Have users actually use the product (develop a customer base)
- Long term goal would be to expand this to multiple universities and establish repeat customers
- Have university students utilize tutoring with less stigma (more or less the ultimate long-term goal)

2.2 Features and Capabilities

- Explain how all of the following fit into the current model, and justify as to why we are being innovative (basically, we are using pieces of what already exist, but consolidating them all into one app)
    - Constraining the user-base strictly to university students
    - Qualifying tutors in real-time based on previously-taken courses
    - Implementing multiple rating systems for both tutors and tutees

    - Providing a toggling availability option for both tutors and tutees
    - Allowing any course at any given university to be offered in-app
    - Implementing open availability and 24/7 scheduling
    - Providing a meaningful notification system
    - Implementing in-app chat
    - Giving tutors the ability to deny meeting requests
    - Automatically calculating competitive pay-rates based on market parameters
    - Allowing for both in-person and online meetings
    - Providing a mechanism for automatic payments within the app.

2.3 Major Functional Components

- Display the MFCD here and touch on what classifies an entity, hardware, and software.
- Touch on why it is put together the way it is (elaborate on the flow).
- Figures
    - 2

2.3.1 Software

- GUI
- User Account
- Network
- Database (Firebase)
- Parsing Algorithm
- Google Calendar API
- Braintree API (payments)
- Google Maps API

2.3.2 Hardware

- Android smartphones

2.3.3 Entities

- University students
    - Tutors
    - Tutees
- University Registrar
- Student email
- User G Suite

3. Identification of Case Study

- Current projected customers
    - Current ODU students seeking tutoring
    - Current ODU students seeking to tutor
        - Private tutors looking to expand their client base
        - Students looking to make extra money by tutoring when they have some free time

- A centralized platform for all tutoring needs would greatly benefit the school in which the app is located, and the students using the app.
    - Pass/Fail/Withdraw rates could decrease because more tutoring = more understanding = better grades (reference presentations for the exact source reference)
    - Students save their grades/money
- Future projected customers
    - Students seeking tutoring advertisement/services on other college campuses

4 Tutor Dash Prototype Description

- Touch on what is going to be in this section (RWP vs. prototype, UI/UX, Firebase)
- How will our prototype differ from the RWP?
- Tables
    - 1
    - There will be a lot of mocked up data (View the RWP vs. prototype figure).
    - No blacklists
    - No appeals for ratings (meaning that no one can contact Tutor Dash directly and say “I want you to investigate this incident because it is affecting me negatively”).
    - No deposits on sessions (threat of poor ratings will deter users from cancelling or not showing up)
    - We won’t account for users lying about being in meetings when they aren’t (e.g., The tutee doesn’t show up to a meeting, but says he/she is there, and he/she also says that the tutor is there. Meanwhile, the tutor says the tutee didn’t show up...so what happens regarding ratings and payments? We are not accounting for it). Solving these issues requires human resources.
- 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 for applying tutors, the rating system, and the utilization of existing APIs to handle tutoring sessions via web conferencing.
    - Essentially, we consolidate everything that already exists, but we add new features like pay-rate calculation and make automating meeting setup/payments much more convenient.

4.1 Prototype Architecture

- Use this section as a way of introducing the big picture for the prototype (everything from the RWP MFCD stays except we are just using ODU for university registrar).
- Make sure to reference the RWP MFCD when you discuss this.

4.1.1 Prototype Hardware

- End user Android phone

4.1.2 Prototype Software

- Firebase database server
    - Will hold all user data necessary for major functions
- Android OS
    - Will run the app for the user
- Braintree

        - Will process payment for both parties
    - Algorithms
        - Will leverage several APIs, including Google Calendar API, Google Maps API, and various Firebase APIs
        - Touch on all of these and explain how they will work in the context of the software as a whole
            - PDF transcript parser
            - Relative distance calculator
            - Pay-rate calculator
            - Web-conference creator
            - Payment logic

4.2 Critical Design Components

- We have three components that essentially make up the entire app (reference the design presentation for help). Summarize all of them and explain their significance as a whole to the product.
    - Database (Firebase) - acting as our network/server
    - UI/UX - serves as the front-end to deliver content in a manageable and clear way
    - Algorithms - serve as the back-end, and will handle most of the app's functionality (these are what really make it unique).

4.2.1 Database

- Firebase
    - No SQL, JSON, document-based database
    - Cloud storage (Team member has working experience with Firebase)
    - Scalability
    - Easy-to-use APIs designed for android apps
    - Explain the schema
- Tables
    - 2

4.2.2 UI/UX

- 4 phases
    - Login/signup
    - Tutor/tutee discovery
    - Session selection
    - Active session
- 2 views
    - Settings/options
    - User profile
- Figures
    - 3.1, 3.2, 3.3, 3.4, 3.5.1, 3.5.2, 3.6, 3.7

4.2.3 Algorithms

- Summarize what each of these are doing and why they are necessary. Reference the website and designs for purpose, tools, and parameters.

        - PDF transcript parser
        - Relative distance calculator
        - Pay-rate calculator
        - Web-conference creator
        - Payment logic
    - Figures
        - 4.1, 4.2, 4.3, 4.4, 4.5

4.3 Prototype Features and Capabilities

- What does the prototype demonstrate?
    - The prototype will demonstrate a user friendly application that will allow customers to connect to their respective counterpart.
    - This will include a simple to use payment method.
    - An intuitive UI.
    - Simple transcript input portion to prove knowledge base.
    - Simple connection to a Google Hangout meeting and/or a specific location meet up.
    - User rating system.
    - A chat windows for both tutor and tutee to communicate.
- Why is that significant in showing how the problem is solved?
    - The objective is to create a centralized, understandable platform so that there are no frictions when an individual is looking for tutoring help. If the application has an intuitive design, simple tutor/tutee finder, and a method of spontaneous meetup, then the objective will be achieved.
- How have you demonstrated success?
    - We will demonstrate success by providing a 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.
    - RWP and Prototype will differ, as there are fewer requirements for a working product (add reference to figure here…)

4.4 Risks and Mitigations

- How does the prototype address the CS 410 project risk mitigation?
    - The rating system will address poor tutees/tutors.
        - This encompasses user abuses.
    - PDF transcript parser will mitigate tutors faking qualifications
    - Tutor/tutee leaves a false negative review
        - This will be balanced out by using averages on the rating system, as well as allowing users to challenge false reviews.
    - User does not show up for specific meeting.
        - We will preallocate payments using braintree.
    - Users will try to book overlapping sessions.
        - Google Calendar API will mitigate this issue.
    - Tutors does not have the hardware to adequately using web conferencing.
        - The rating system will resolve this issue.
- Figures
    - 5.1, 5.2

4.5 Prototype Objectives and Goals

- Describe the functional goals and objectives.
    - Clean UI/ease of use.
    - Find a tutee/tutor in a timely manner.
    - Proper payment system.
    - Proper connection to a medium for the tutoring session.
    - Rating system.

4.6 Prototype Development Challenges

- Most team members are inexperienced in Android app development
- Implementing web conferencing algorithm (permissions issues dealing with other user's G Suites)
- Working with Braintree API that handles payments
- Connecting the user interface to the backend algorithms and database
- Receiving messages between users in real time
- Handling of funds between tutees and tutors
- Handshake agreement to initialize start of meeting
- Getting alerted about potential compatible tutors/tutees
- Receiving a competitive rate accurately calculated based on a variety of factors
- Testing the user experience
- Development time constraints

5 Glossary

- **Course-specific tutoring**: Academic assistance services provided for a particular course at a particular university.
- **DFWI**: An acronym for Drop/Fail/Withdraw/Incomplete. This relates to university course incompletion status.
- **DFWI rates**: Represents the ratio of university students who do not complete their courses to students who do complete their courses.
- **Direct Competitor**: Another product or company which is solely involved in the same domain space as Tutor Dash.
- **Entity**: A person, object, or external server that serves as a leveraged functional component of the Tutor Dash product.
- **FERPA**: The Family Educational Rights and Privacy Act is a United States federal law that protects the privacy of educational records.
- **Indirect Competitor**: Another product or company which is involved in the same domain space as Tutor Dash, but doesn't not focus solely in that space.
- **Tester**: Individuals responsible for testing the quality of the software.
- **Tutee**: A university student seeking academic assistance.
- **Tutor**: A university student offering independent tutoring services that are qualified based off previously taken courses.
- **Uber**: A ride-hailing company that offers the Uber mobile app, which you can use to submit a trip request that is automatically sent to an Uber driver near to you, alerting the driver to your location.

6 References

1. "Academic Tutoring in Comprehensive Universities." Hanover Research, 2014. URL: https://www.hanoverresearch.com/wp-content/uploads/2017/08/Academic-Tutoring-in-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_authors_&utm_medium=Email&utm_content=T1_11.5e1_author&utm_campaign=Email_publication&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#AN=114789419&db=ehh
13. Qayyum, Adnan. "Student Help-Seeking Attitudes and Behaviors in a Digital Era." International Journal of Educational Technology in Higher Education, vol. 15, no. 1, 2018, doi: 10.1186/s41239-018-0100-7. URL: https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-018-0100-7
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