*An application based on linking qualified tutors with students who need help and are tired of having to branch out on their own to find quality assistance.*

*TutorHub*

Project Portfolio

*[9/16]*



[**Introduction**](#_heading=h.gjdgxs) **3-4**

[**The Tutorhub Team**](#_heading=h.30j0zll) **4**

[**System Requirements**](#_heading=h.1fob9te) **4**

[Requirements](#_heading=h.3znysh7) 4-5

[User Stories](#_heading=h.3dy6vkm) 5

*User Story #1*

*User Story #2*

*User Story #3*

[**Project Management**](#_heading=h.1t3h5sf) 5

[Continuity of Operations Plan (COOP)](#_heading=h.4d34og8) 5

[Project Plan](#_heading=h.2s8eyo1) 6

[*System Architecture Design and Development*](#_heading=h.17dp8vu) 6

[*System Implementation <Milestone 2: Architecture & Milestone 3: System Implementation> 0*](#_heading=h.3rdcrjn)

[Project Postmortem <Postmortem> 0](#_heading=h.26in1rg)

[*Project Wins 0*](#_heading=h.lnxbz9)

[*Root Cause Analysis 0*](#_heading=h.35nkun2)

[*Lessons Learned 0*](#_heading=h.1ksv4uv)

[**System Design 0**](#_heading=h.44sinio)

[System Architecture <Milestone #2: System Architecture> 0](#_heading=h.2jxsxqh)

[*Component Design 0*](#_heading=h.z337ya)

[*Data Flow 0*](#_heading=h.3j2qqm3)

[System Components <Milestone 3: System Implementation> 0](#_heading=h.1y810tw)

[*Component [Component Name 1] 0*](#_heading=h.4i7ojhp)

[*Component [Component Name 2] 0*](#_heading=h.2xcytpi)

[*Component [Component Name n] 0*](#_heading=h.1ci93xb)

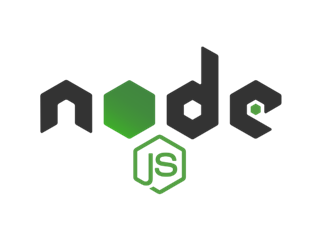
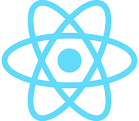
[Design Pattern <Milestone 3: System Implementation>](#_heading=h.3whwml4) 0

[Design Pattern <Milestone 3: System Implementation>](#_heading=h.2bn6wsx) 0

# Introduction

LSU courses, especially the courses required for a student's major, can be challenging to understand, comprehend, and succeed in. Even though professors are there to help out their students, they usually do not understand the students perspective of learning and their thought process. To get hands on help with their schoolwork students will usually seek out tutors and people who have already taken the subject who can use their personal experiences of the course to help the student succeed in the course. The problem with getting a tutor is first navigating through a site that helps students and tutors reach out to one another. The second problem is the finding the availability and preferences of both student and tutor.

Tutorhub can be the first solution for tutors and students to easily access a site and quickly find help from tutors only at LSU. Tutorhub can be used on Iphones or computers and will have many options of tutors located in different categories of our system. Students would just have to login, browse through the system for the specific tutor and schedule a time to meet up via online or in person. But not only will Tutorhub just be for locating tutors but also a site that provides handwritten notes from the tutors that took the class so that students can be given former-student notes.







Core Features:

* Profiles made by users denoting whether they are a tutor or a student and what they are looking for
* A messaging system between users to meet each other, communicate prices, set a meeting location, or schedule zoom meeting
* A payment system that enables users to pay tutors through the app via Plaid

Viable Features:

* A rating system for tutors that lets users know if a tutor is the right fit for them
* Allow users to connect with others from their university if signed up with a university email I.E email@lsu.edu
* A feedback feature to help the tutorhub team deal with any systemic problems

Stretch Features

* Implement user creation through google, outlook, and apple accounts
* Background checks for tutors wishing to meet others in person

# TutorHub Team Structure

Brandon – Head of Back-end development, system architect

Tyler – Head of Front-end development, project manager

Simon – Full stack developer

Brandt – Front-end development, graphic artist

Troy – Back-end development, scrum master

Logan – Back-end development, document management

Github: https://github.com/brandvdo/TutorHub

# System Requirements

## Requirements

* Smartphone running
  + IOS 16
  + Android Marshmallow 6.0 or later
* Web Browser
  + Windows 7, Windows 8, Windows 8.1, Windows 10 or later
    - An Intel Pentium 4 processor or later that’s SSE3 capable
  + MAC
    - macOS High Sierra 10.13 or later
  + Linux
    - 64-bit Ubuntu 18.04+, Debian 10+, openSUSE 15.2+, or Fedora Linux 32+
    - An Intel Pentium 4 processor or later that’s SSE3 capable

## User Stories

* As a struggling student, I want to find a tutor near me, so I can learn a subject more efficiently.
* As an experienced mathematician, I want to find someone in need of math tutoring, so I can make some extra money.
* As a busy student, I want to see the qualifications of my tutor, so I can avoid wasting my time.

# Project Management

## Continuity of Operations Plan (COOP)

Our group’s main method of communication is a dedicated discord server where we will be coordinating our efforts and ensuring that no member is unaware of their next step. This server will also be used to schedule in person programming sessions with the members. We also have made a group text with all our phone numbers in case any of us need immediate assistance or need to get in touch with one of our members as soon as possible. In the case of one of our members being sick or unable to meet in person, we plan to use audio chats through discord to still meet. In the case that a visual medium is required, we plan to use zoom and screen sharing so as to not fall behind on our meetings. If one of our members were to drop out of the course or not be able to complete their duties, their remaining workload would be distributed equally amongst the remaining members, based on personal preference

## Project Plan

### System Architecture Design and Development

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Activity** | **Pre #** | **Estimated**  **Effort** | **Actual**  **Effort** | **Estimated**  **Start Date** | **Estimated**  **Finish Date** | **Actual**  **Start Date** | **Actual**  **Finish Date** |
| 1 | User account system |  | 10-15h |  | 9/12 | 9/30 | 9/12 | 9/25 |
| 2 | Core app interface |  | 10-15h |  | 9/16 | 9/30 |  |  |
| 3 | User profiles | 1,2 | 6-10h |  | 9/31 | 10/4 |  |  |
| 4 | User messaging | 3 | 4-6h |  | 10/5 | 10/15 |  |  |
| 5 | Payment System | 3 | 5-10h |  | 10/5 | 10/25 |  |  |
| 6 | Tutor feedback | 3 | 3-5h |  | 10/30 | 11/3 |  |  |

### System Implementation <Milestone 2: Architecture & Milestone 3: System Implementation>

[Milestone 2 (Architecture): The Project Plan WBS provides a list of activities/tasks to be undertaken to complete Milestone 3 (System Implementation). The WBS activity chart should include task dependencies, estimated level of effort, and expected start and completion dates.

Milestone 3 (System Implementation): The WBS activity chart for the milestone should be updated to include actual level of effort and start and completion dates.]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Activity** | **Pre #** | **Estimated**  **Effort** | **Actual**  **Effort** | **Estimated**  **Start Date** | **Estimated**  **Finish Date** | **Actual**  **Start Date** | **Actual**  **Finish Date** |
|  |  |  |  |  |  |  |  |  |

## Project Postmortem <Postmortem>

### Project Wins

[Provide a bulleted list of at least 3 positive aspects of the project.]

### Root Cause Analysis

[Provide a bulleted list of at least 3 negative aspects of the project. For each negative, provide the answer to the three successive “Why” questions. ]

### Lessons Learned

[For each negative aspect identified in the Root Cause Analysis, provide a mitigation strategy (i.e., what process should be introduced) to ensure that the problem is not repeated in subsequent projects.]

# System Design

[*Include a short (1-2 sentences) statement about system design*.]

## System Architecture <Milestone #2: System Architecture>

[*A short description of the system architecture.*]

### Component Design

[*Insert image of system architecture component diagram.*]

[*Architecture overview, to include user I/O, external data sources, and major system components.*]

### Data Flow

[*Insert image of system architecture data flow diagram.*]

[*Architecture data flow discussion: a high-level description of the data between both internal major components and external data sources.*]

## System Components <Milestone 3: System Implementation>

[*Include a component sub-section for each component in the architecture diagram. Each component subsection will include a class diagram*]

### Component [Component Name 1]

[*A short description of the component*.]

[*An EA class diagram of the component that includes method parameters.*]

### Component [Component Name 2]

[*A short description of the component*.]

[*An EA class diagram of the component that includes method parameters.*]

### Component [Component Name n]

[*A short description of the component*.]

[*An EA class diagram of the component that includes method parameters.*]

## Design Pattern <Milestone 3: System Implementation>

[*Class diagram of design pattern incorporated into the project. Pattern must be specific to the project and not a general design pattern class diagram. The project must include at least 1 design pattern covered in class.*]

## Design Pattern <Milestone 3: System Implementation>

[*Class diagram of design pattern incorporated into the project. Pattern must be specific to the project and not a general design pattern class diagram. A second design pattern may be included for bonus points.*]