



# Design Document

---

## TutorFIT

A Mobile Application Connecting Students and Tutors at the  
Florida Institute of Technology

Project Team:

- Eleanor Barry (ebarry2022@my.fit.edu)
- Sidney Nedd (snedd2020@my.fit.edu)
- Samaher Damanhori (sdamanhori2020@my.fit.edu)

Faculty Advisor:

Dr. Khaled Salhoub (ksalhoub@fit.edu)

Client:

- Dr. Khaled Salhoub
- Students of the Florida Institute of Technology

10/02/2023



# Contents

<b>1. Introduction.....</b>	<b>3</b>
1.1 Purpose.....	3
1.2 Scope.....	3
1.3 Overview of TutorFIT Mobile Application.....	3
<b>2. System Architecture.....</b>	<b>4</b>
2.1 tutorFIT UML Diagram.....	4
<b>3. Module Descriptions.....</b>	<b>5</b>
3.1. User Registration Module.....	5
3.2. Scheduling Module.....	5
3.3. Communication Module.....	5
3.4. Push Notifications/Alerts Module.....	5
3.5. Student Reviews and Ratings Module.....	5
3.6. Event Tracking Module.....	6
3.7. User Engagement and Retention Module.....	6
<b>4. Graphical User Interface (GUI) Design.....</b>	<b>6</b>
4.1. Login Screen.....	6
4.2. Student Dashboard.....	6
4.3. Tutor Dashboard.....	6
4.4. Messaging Interface.....	6
<b>5. Additional Design.....</b>	<b>7</b>
5.1. Algorithms (Pseudocode).....	7
5.1.1 User Registration Algorithm.....	7
5.1.2 Scheduling Algorithm.....	7
5.1.3 Communication Algorithm.....	8
5.1.4 Push Notifications/Alerts Algorithm.....	8
5.1.5 Student Ratings and Reviews Algorithm.....	9
5.1.6 Event Tracking Algorithm.....	9
5.1.7 User Engagement and Retention Algorithm.....	9
5.2. Database Design.....	9
<b>6. Conclusion.....</b>	<b>10</b>

# 1. Introduction

## 1.1 Purpose

The purpose of this design document is to provide a comprehensive blueprint for the development of the TutorFIT mobile application. This document outlines the architectural, user interface, and implementation details that will guide the development process. It serves as a reference for the development team, ensuring that the application aligns with the specified requirements and objectives.

## 1.2 Scope

The scope of this design document encompasses the entire architecture and design of the TutorFIT mobile application. This includes defining the software and hardware components, the user interface layout, system behavior, and data flow. It also considers the integration of third-party tools and libraries required for the application's functionality.

## 1.3 Overview of TutorFIT Mobile Application

The TutorFIT mobile application is a dynamic platform designed to connect students and tutors at the Florida Institute of Technology (FIT). Its primary goal is to streamline the process of connecting students with accessible tutoring resources while enhancing the overall learning experience within the FIT student community.

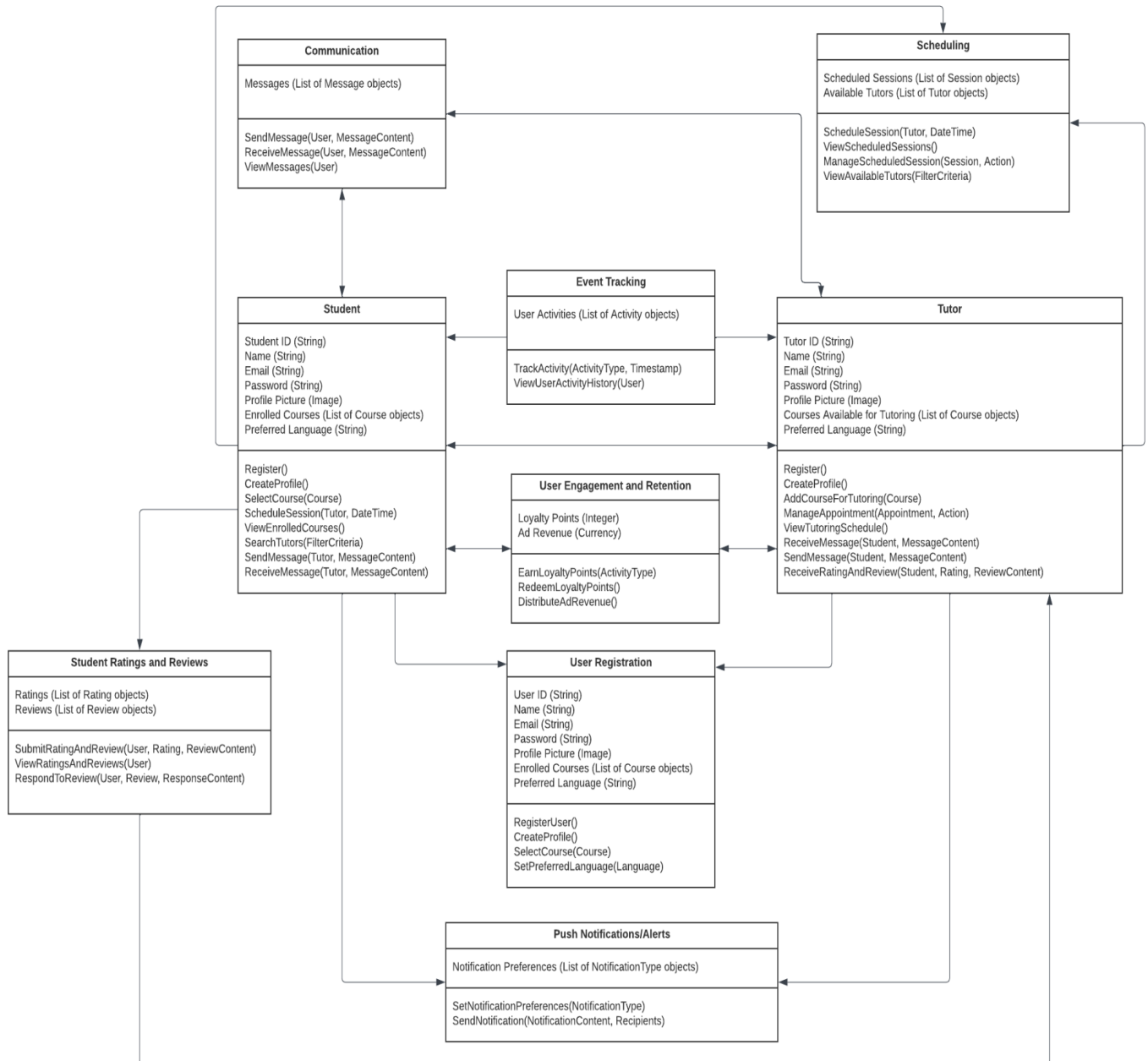
### Key Objectives:

- Simplify user registration and profile creation for students and tutors.
- Facilitate scheduling of tutoring sessions with available tutors.
- Enable real-time communication between students and tutors.
- Implement push notifications and alerts for timely updates.
- Support student reviews and ratings for tutors.
- Track user engagement and interactions within the app.
- Foster user engagement and retention through a loyalty program and ad revenue redistribution.

The design document elaborates on these objectives, providing detailed architectural plans and design considerations to achieve them effectively.

## 2. System Architecture

### 2.1 tutorFIT UML Diagram



## 3. Module Descriptions

### 3.1. User Registration Module

- RegisterUser(): Registers a new user (student or tutor) with the application.
- CreateProfile(): Allows users to create or update their profile information.
- SelectCourse(Course): Adds a course to the user's list of enrolled courses.
- SetPreferredLanguage(Language): Sets the user's preferred teaching or learning language.

### 3.2. Scheduling Module

- ScheduleSession(Tutor, DateTime): Allows students to schedule tutoring sessions with available tutors.
- ViewScheduledSessions(): Displays the user's scheduled tutoring sessions.
- ManageScheduledSession(Session, Action): Allows users to manage scheduled sessions (e.g., cancel a session).
- ViewAvailableTutors(FilterCriteria): Shows a list of available tutors based on filter criteria.

### 3.3. Communication Module

- SendMessage(User, MessageContent): Allows users to send messages to other users.
- ReceiveMessage(User, MessageContent): Receives messages from other users.
- ViewMessages(User): Displays the user's messages and conversations.

### 3.4. Push Notifications/Alerts Module

- SetNotificationPreferences(NotificationType): Allows users to configure their notification preferences.
- SendNotification(NotificationContent, Recipients): Sends push notifications or alerts to users.

### 3.5. Student Reviews and Ratings Module

- SubmitRatingAndReview(User, Rating, ReviewContent): Allows students to submit ratings and reviews for tutors.
- ViewRatingsAndReviews(User): Displays ratings and reviews received by users.
- RespondToReview(User, Review, ResponseContent): Enables tutors to respond to student reviews.

### 3.6. Event Tracking Module

- `TrackActivity(ActivityType, Timestamp)`: Records user activities (e.g., sign-ups, appointment bookings).
- `ViewUserActivityHistory(User)`: Displays the user's activity history.

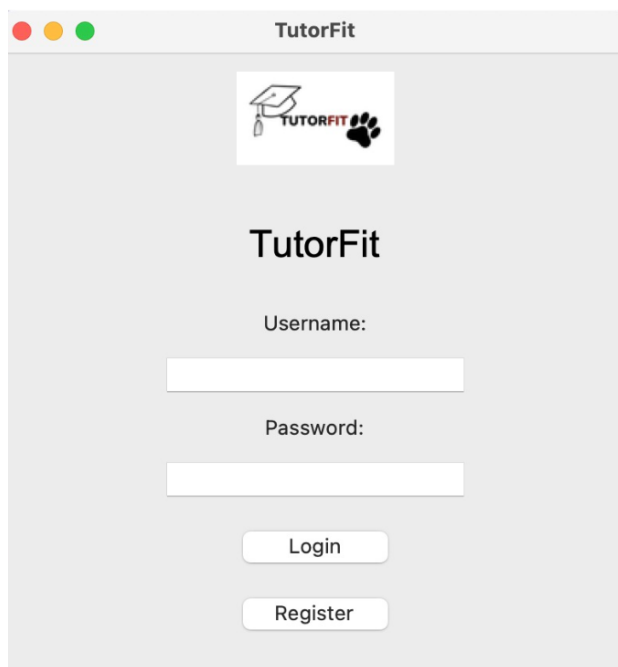
### 3.7. User Engagement and Retention Module

- `EarnLoyaltyPoints(ActivityType)`: Rewards users with loyalty points for engaging in specific activities.
- `RedeemLoyaltyPoints()`: Allows users to redeem loyalty points for free tutoring sessions.
- `DistributeAdRevenue()`: Distributes ad revenue to dedicated tutors as additional income.

## 4. Graphical User Interface (GUI) Design

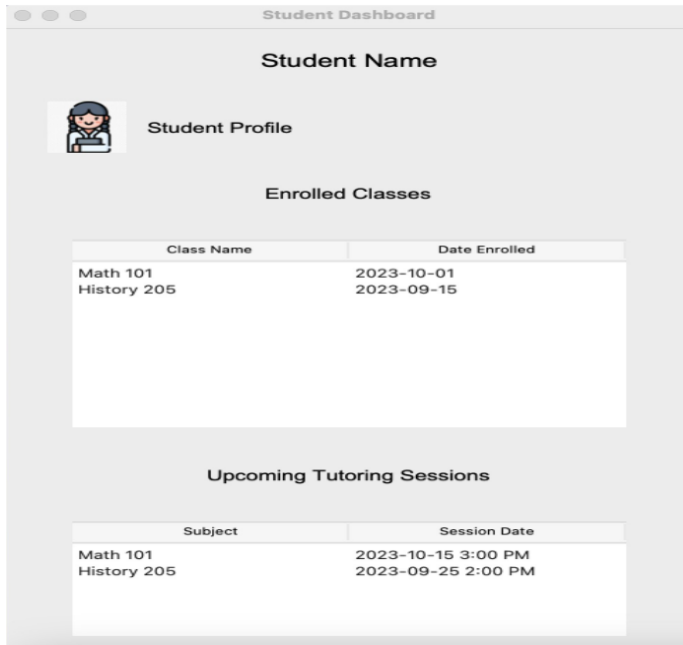
Provide sketches (mock-ups) of different screens to illustrate key functionalities. Include descriptions for each screen.

### 4.1. Login Screen



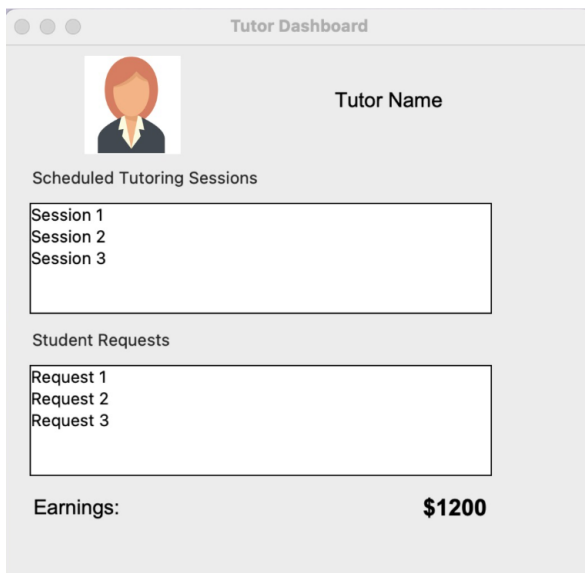
Description: This interface provides a dedicated portal where users can enter their username and password to access their accounts

## 4.2. Student Dashboard



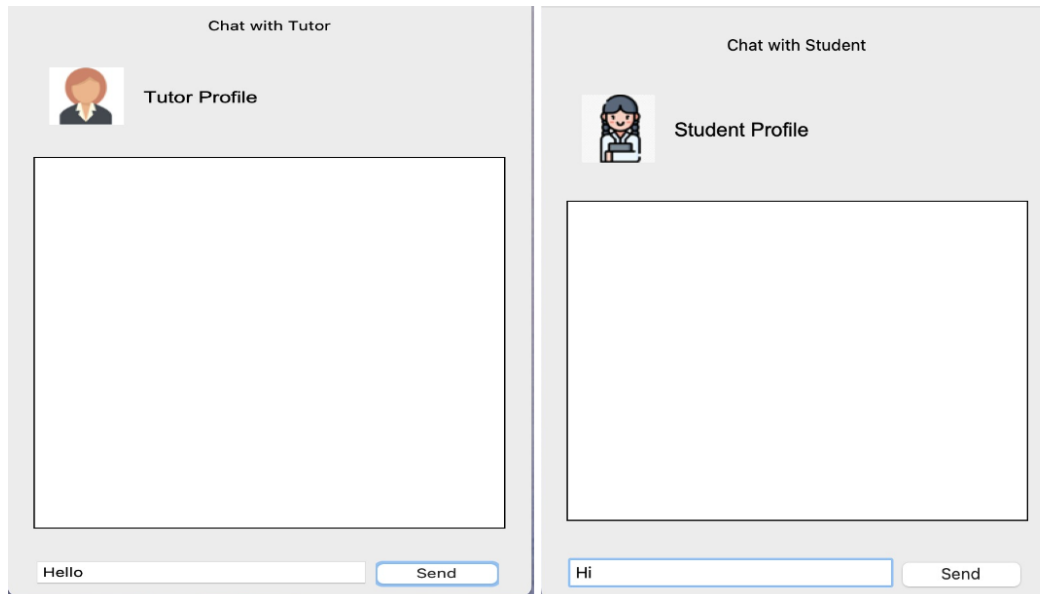
Description: The student dashboard serves as a comprehensive information panel, presenting details on enrolled classes, including class names and dates, upcoming tutoring sessions, and additional pertinent information.

## 4.3. Tutor Dashboard



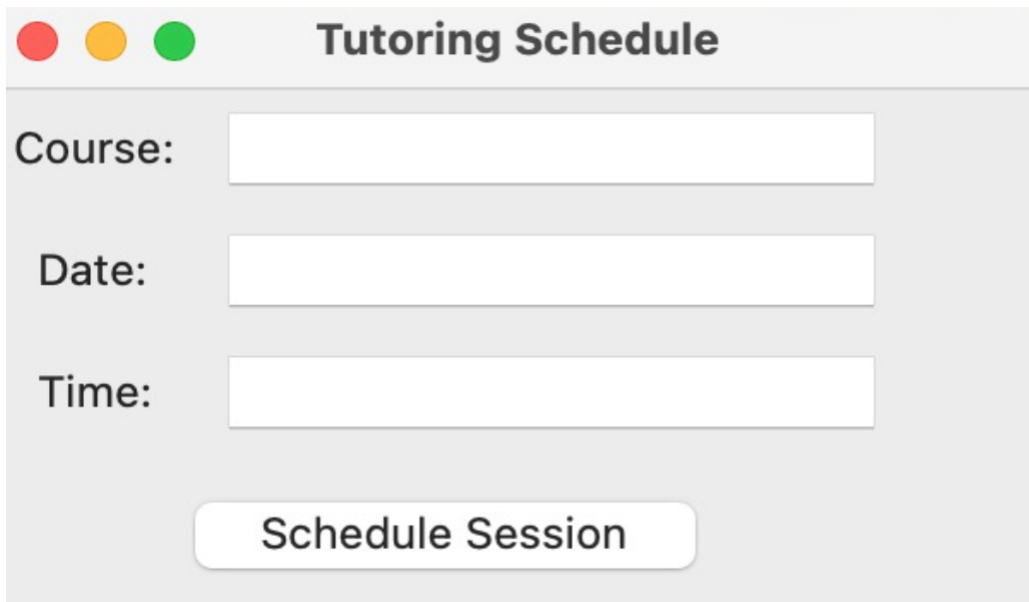
Description: The tutor dashboard provides a detailed overview of upcoming scheduled tutoring sessions, student requests, and accumulated earnings.

## 4.4. Messaging Interface



Description: This platform enables users to exchange messages instantly and seamlessly.

## 4.5. Tutoring scheduling

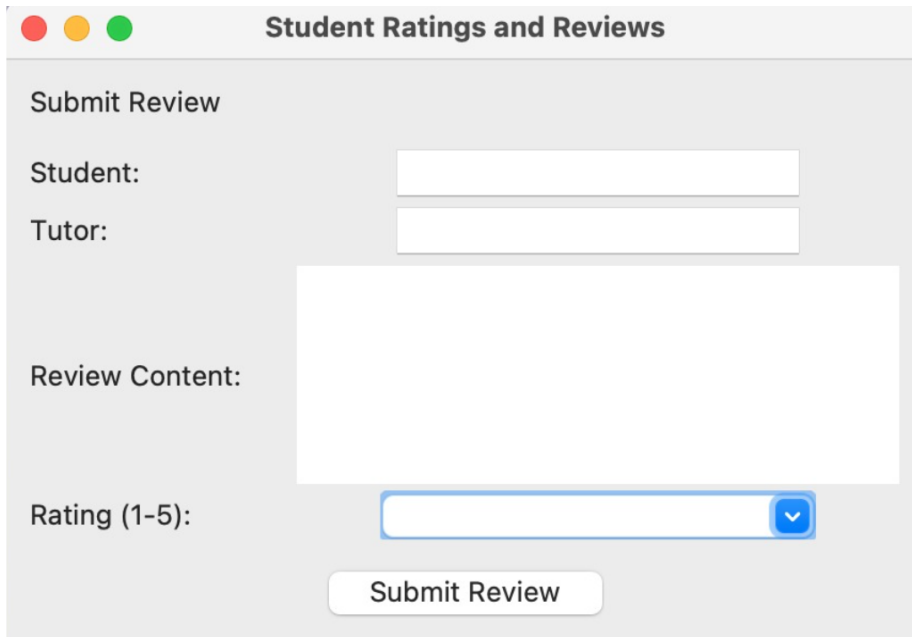


The image shows a "Tutoring Schedule" form with a title bar containing three colored circles (red, yellow, green) and the text "Tutoring Schedule". The form contains three input fields: "Course:", "Date:", and "Time:". Below these fields is a "Schedule Session" button.

Description: This platform facilitates students in arranging tutoring sessions, allowing them to select the desired course, date, and time conveniently.



## 4.6. Student rating and Reviews



Description: On this platform, students have the opportunity to evaluate their tutoring experiences. By entering their name, the tutor's name, and their feedback, they can assign a rating ranging from 1 (lowest) to 5 (highest), offering tutors valuable insights into their performance.

## 5. Additional Design

### 5.1. Algorithms (Pseudocode)

#### 5.1.1 User Registration Algorithm

```
function userSignUp(name, email, password):
    if email is not already registered:
        create a new user record with name, email, and hashed
password
        add the user record to the database
        return success message
    else:
        return error message indicating email is already
registered
```

```
function createOrUpdateUserProfile(user, profileDetails):
    update the user's profile record with provided details
(e.g., name, contact info, profile picture)
```

```
return success message

function selectCourses(user, selectedCourses):
    if user is a student:
        add selected courses to the user's enrolled courses list
    if user is a tutor:
        add selected courses to the user's list of courses they
can tutor
    return success message

function setUserLanguagePreference(user, language):
    set the user's preferred teaching or learning language
    return success message
```

### 5.1.2 Scheduling Algorithm

```
function scheduleTutoringSession(student, tutor, course, date,
time):
    if student is enrolled in course and course is in tutor's
list of available courses:
        create a new session record with student, tutor, course,
date, and time
        add the session to the database
        return success message
    else:
        return error message indicating scheduling is not
possible

function viewScheduledSessions(user):
    sessions = query database for sessions where user is the
student or tutor
    if sessions is not empty:
        return sessions
    else:
        return no sessions found message
```

### 5.1.3 Communication Algorithm

```
function sendMessage(sender, receiver, messageContent):
    create a new message record with sender, receiver, and
message content
    add the message record to the database
    return success message
```

```
function receiveMessages(user):  
    messages = query database for messages where user is the  
sender or receiver  
    return messages
```

#### 5.1.4 Push Notifications/Alerts Algorithm

```
function sendPushNotification(user, notificationContent):  
    create a new notification record with user and notification  
content  
    send a push notification to the user's device  
    add the notification record to the database  
    return success message
```

```
function receiveNotifications(user):  
    notifications = query database for notifications related to  
the user  
    return notifications
```

#### 5.1.5 Student Ratings and Reviews Algorithm

```
function submitReview(student, tutor, reviewContent, rating):  
    create a new review record with student, tutor, review  
content, and rating  
    add the review record to the database  
    return success message
```

```
function viewReviews(tutor):  
    reviews = query database for reviews where tutor is the  
reviewed tutor  
    return reviews
```

#### 5.1.6 Event Tracking Algorithm

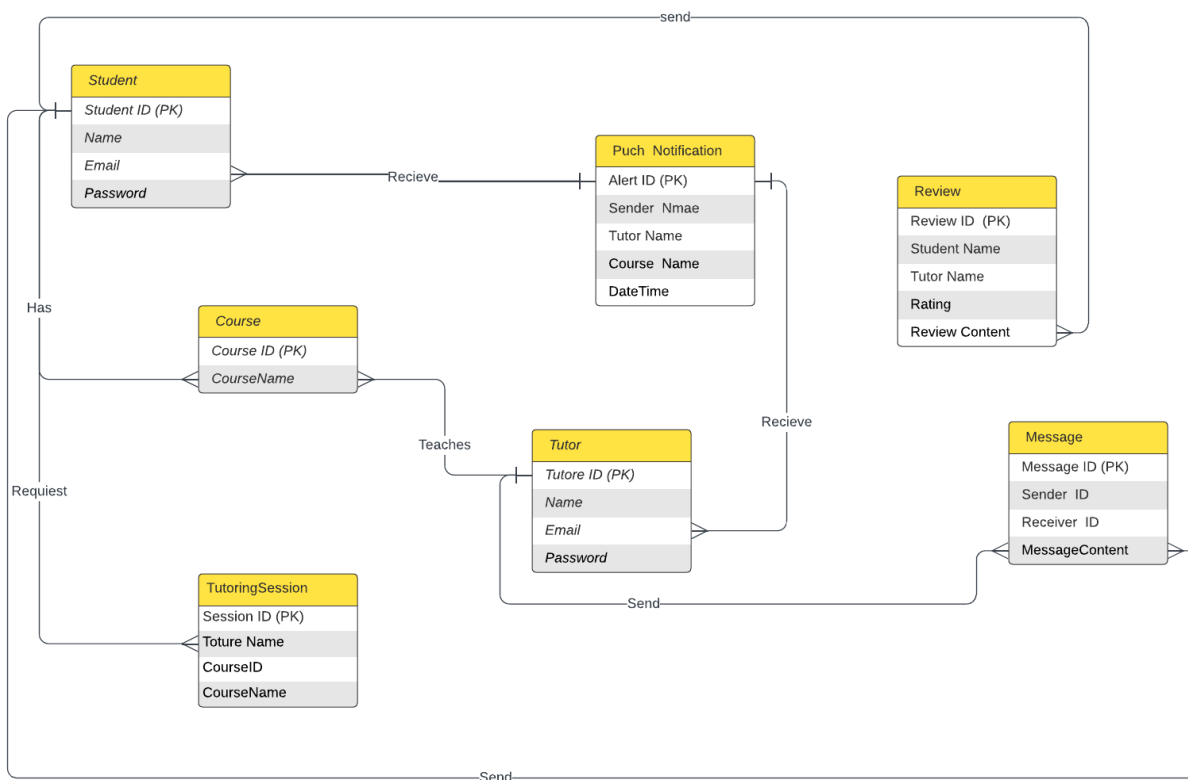
```
function trackUserActivity(user, activityType):  
    create a new activity record with user and activity type  
    add the activity record to the database
```

### 5.1.7 User Engagement and Retention Algorithm

```
function provideLoyaltyReward(user, rewardType):
    if user qualifies for a loyalty reward:
        provide the user with the specified reward (e.g., free tutoring session)
    return success message
```

```
function distributeAdRevenue(tutor):
    if tutor qualifies for ad revenue redistribution:
        distribute ad revenue earnings to the tutor
    return success message
```

### 5.2. Database Design



1. Student:
  - o Description: This table stores student information.

- Attributes:
    - Student ID
    - Name
    - Email
    - Password
  - Primary Key: Student ID
2. Tutor:
- Description: This table stores information related to the tutoring process.
  - Attributes:
    - Tutor ID
    - Name
    - Email
    - Password
  - Primary Key: Tutor ID
3. Tutoring Session:
- Description: This table stores specific data about each tutoring session.
  - Attributes:
    - Session ID
    - Tutor Name
    - Student Name
    - DateTime
    - CourseName
  - Primary Key: Session ID
4. Course:
- Description: This table stores course-related information.
  - Attributes:
    - Course ID
    - CourseName
  - Primary Key: Course ID
5. Push Notification:
- Description: This table stores the information of the push notifications sent to users.
  - Attributes:
    - Alert ID
    - MessageContent
    - DateTime
  - Primary Key: Alert ID
6. Message:

- Description: This table stores messages exchanged between students and tutors .
- Attributes:
  - Message ID
  - Sender Name
  - Sender ID
  - Receiver ID
  - MessageContent
  - DateTime
- Primary Key: Message ID

#### 7. Review:

- Description: This table stores reviews given by students about their tutoring sessions.
- Attributes:
  - Review ID
  - Student Name
  - Tutor Name
  - Rating
  - Review Content
- Primary Key: Review ID

## 6. Conclusion

In conclusion, this design documentation delineates a thorough framework for the development of the TutorFIT mobile application, intended to bridge the gap between students and tutors at the Florida Institute of Technology. It meticulously details the app's architecture, the functionalities of different modules, the design of the Graphical User Interface (GUI), pertinent algorithms, and the structure of the database to steer the developmental trajectory. The overarching objective of TutorFIT is to streamline the process of tutoring, elevate the level of communication, and boost user interaction, thereby refining the learning experience for students at FIT.