



Test Plan

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

| | |
|---|-----------|
| 1. Introduction..... | 3 |
| 1.1 Purpose of the Document..... | 3 |
| 1.2 Scope of Testing..... | 3 |
| 1.3 Objective..... | 3 |
| 1.4 Test Environment..... | 3 |
| 1.5 References..... | 3 |
| 2. Features to be Tested..... | 4 |
| 2.1 User Registration..... | 4 |
| 2.2 Scheduling..... | 4 |
| 2.3 Communication..... | 4 |
| 2.4 Push Notifications/Alerts..... | 5 |
| 2.5 Student Reviews and Ratings..... | 5 |
| 2.6 Event Tracking..... | 5 |
| 2.7 User Engagement and Retention..... | 5 |
| 3. Testing Approach..... | 6 |
| 3.1 Testing Types..... | 6 |
| 3.2 Test Deliverables..... | 7 |
| 3.3 Testing Schedule..... | 8 |
| 4. Test Cases..... | 8 |
| 4.1 User Registration..... | 8 |
| 4.2 Scheduling..... | 10 |
| 4.3 Communication..... | 12 |
| 4.4 Push Notifications/Alerts..... | 13 |
| 4.5 Student Reviews and Ratings..... | 13 |
| 4.6 Event Tracking..... | 14 |
| 4.7 User Engagements and Retention..... | 15 |
| 5. Testing Risks..... | 15 |
| 6. Testing Resources..... | 16 |
| 6.1 Testing Team..... | 16 |
| 6.2 Hardware Resources..... | 16 |
| 6.3 Software Resources..... | 16 |
| 6.4 Test Data..... | 16 |
| 6.5 Test Environment..... | 16 |
| 7. Test Execution..... | 17 |
| 8. Reporting..... | 17 |
| 9. Conclusion..... | 17 |

1. Introduction

1.1 Purpose of the Document

The purpose of this document is to outline the test plan for the TutorFIT mobile application, which aims to connect students and tutors at the Florida Institute of Technology (FIT).

1.2 Scope of Testing

This test plan covers the testing of various features and functionalities of the TutorFIT mobile application, including:

1. User Registration
2. Scheduling
3. Communication
4. Push Notifications/Alerts
5. Student Reviews and Ratings
6. Event Tracking
7. User Engagement and Retention

1.3 Objective

The objective of testing is to ensure the functionality, usability, and reliability of the TutorFIT mobile application, as well as to identify and address any defects or issues.

1.4 Test Environment

1. Mobile Devices: iOS and Android smartphones and tablets
2. Operating Systems: iOS, Android
3. Test Management Tool: [Specify when applicable]
4. Test Automation Tool: [Specify when applicable]

1.5 References

Project Plan for TutorFIT Mobile Application.

2. Features to be Tested

2.1 User Registration

1. User Sign-up: Testing the registration process where users can register by providing their name, email address, and password.
2. User Profiles: Testing the creation and editing of user profiles, including personal information such as name, contact details, and a profile picture.
3. Course Registration: Testing the ability of users to select and register for specific courses from a predefined list.
4. Live Search: Testing the live search feature during registration, allowing users to quickly find and select their courses, professors, and departments.
5. Language Preference: Testing the feature that allows users to set their preferred teaching or learning language during registration.

2.2 Scheduling

1. Class Enrollment: Testing whether students can view the list of classes they are enrolled in.
2. Tutor Selection: Testing the functionality that allows students to access information about available tutors for their classes.
3. Tutor Scheduling: Testing the process of scheduling tutoring sessions with available tutors.
4. Tutor Management: Testing whether tutors can view the list of classes they have chosen to tutor for during registration.
5. Tutor Requests: Testing the ability of tutors to manage incoming tutoring requests from students, accepting or declining them as needed.
6. Search Filters: Testing the functionality that enables users to search for tutors using filters such as course name, course code, preferred learning language, and department.

2.3 Communication

1. In-App Messaging: Testing the feature that allows users to communicate with each other via an in-app messaging system.
2. Real-time Chat: Testing whether messaging is real-time, leveraging real-time messaging SDKs to ensure immediate communication.
3. Email Integration: Testing the option for users to communicate through email if they prefer email-based communication.

2.4 Push Notifications/Alerts

1. Appointment Alerts: Testing whether users receive push notifications and alerts for appointment updates, including accepted and declined appointments.
2. Reminder Alerts: Testing whether users receive reminders for scheduled tutoring sessions.
3. Message Alerts: Testing whether users are notified of new messages from their tutors.

2.5 Student Reviews and Ratings

1. Review Submission: Testing whether students can submit reviews and ratings after each tutoring session.
2. Review Visibility: Testing whether reviews and ratings are visible to other students considering the same tutor.
3. Tutor Responses: Testing whether tutors can view and respond to student feedback.

2.6 Event Tracking

1. User Engagement Metrics: Testing the integration of a third-party SDK to monitor user engagement, tracking activities such as sign-ups, appointments booked, cancellations, and other relevant interactions.
2. Data Collection: Testing whether the system collects data on user behavior within the app.

2.7 User Engagement and Retention

1. Loyalty Program: Testing whether loyal and frequent users have the advantage of accessing free tutoring sessions.
2. Ad Revenue Redistribution: Testing whether dedicated tutors benefit from ad revenue redistribution, allowing them to earn extra income.
3. Reward System: Testing whether the app encourages users to stay engaged and active through a reward system.

3. Testing Approach

3.1 Testing Types

3.1.1 Functional Testing

Functional testing ensures that the application's features and functionalities work as intended. It verifies that each feature performs the specified functions correctly. This involves both positive and negative testing to ensure robustness

Use Cases: Test the user registration process, scheduling of tutoring sessions, in-app messaging, and other core features for correctness. This may also include boundary value testing and edge cases to ensure comprehensive coverage

3.1.2 Usability Testing

Usability testing evaluates the application's user-friendliness and how easily users can navigate and interact with it. Factors like UI design, clarity of instructions, and error messages are also assessed

Use Cases: Assess how easy it is for users to create profiles, schedule tutoring sessions, and communicate with tutors.

3.1.3 Performance Testing

Performance testing evaluates the application's speed, responsiveness, and scalability under different conditions. This might include stress testing to evaluate the system's behavior under peak loads

Use Cases: Test the application's response times when multiple users are scheduling sessions simultaneously.

3.1.4 Security Testing

Security testing checks for vulnerabilities and ensures that user data is protected from unauthorized access. This can involve penetration testing to identify potential weak spots

Use Cases: Verify that user data is encrypted, and access controls are in place to prevent data breaches.

3.1.5 Compatibility Testing

Compatibility testing ensures that the application functions correctly on different devices, browsers, and operating systems. It's essential to ensure a consistent user experience regardless of the user's device or platform

Use Cases: Test the app on various Android and iOS devices to ensure it works uniformly.

3.2 Test Deliverables

1. **Test Plan:** The test plan document outlines the testing strategy, objectives, scope, schedule, and resources required for testing. It provides a roadmap for testing activities and ensures alignment with project goals.
2. **Test Cases:** Test cases document the detailed steps, input data, and expected outcomes for each test scenario. It guides testers in executing tests and serves as a basis for documenting results.
3. **Test Data:** Test data includes sample input data required for executing test cases. It ensures that testers have the necessary data to perform tests.
4. **Test Execution Reports:** These reports document the results of test case execution, including pass/fail status, defects found, and any deviations from expected outcomes. It provides a record of testing progress and identifies areas that require further attention.
5. **Defect Reports:** Defect reports document identified defects, including their severity, steps to reproduce, and status. It enables the development team to prioritize and address defects.
6. **Test Summary Report:** The test summary report provides an overall summary of testing activities, including a pass/fail summary, testing efforts, and recommendations. It helps stakeholders make informed decisions about the readiness of the application for release.

3.3 Testing Schedule

| Activity | Start Date | End Date | Description |
|---|------------|------------|---|
| Planning & Setting up Testing Environment | 10/02/2023 | 10/08/2023 | Understand the requirements, gather tools, and set up the necessary environments for testing. |
| Unit Testing | 10/09/2023 | 10/13/2023 | Individual units or components of the software will be tested to ensure functionality. |
| Integration Testing | 10/18/2023 | 10/27/2023 | Different software modules are integrated and tested together. |
| System Testing | 10/30/2023 | 11/07/2023 | The full software system is tested for end-to-end functionality. |
| User Acceptance Testing (UAT) | 11/11/2023 | 11/19/2023 | Potential users test the system to verify if it meets their needs. |
| Feedback Integration & Bug Fixes | 11/25/2023 | 12/05/2023 | Incorporate feedback from UAT and rectify identified issues. |
| Performance & Security Testing | 12/06/2023 | 12/10/2023 | Evaluate software under load and assess for potential security threats. |
| Final UAT & Bug Fixes | 12/11/2023 | 12/15/2023 | After initial fixes, another round of UAT and subsequent rectifications. |

4. Test Cases

4.1 User Registration

Test Case 1: User Sign Up

| Test Case ID: TC_UR_001 | |
|--------------------------------|--|
| Description | Verify that a user can successfully register by providing their name, email address, and password. |
| Preconditions | User is on the registration page. |
| Inputs | Name: [Valid Name] Email: [Valid Email] Password: [Valid Password] |
| Expected Output | User is registered successfully. User is redirected to the profile page. User receives a confirmation email. |

Test Case 2: User Profiles

| Test Case ID: TC_UR_002 | |
|--------------------------------|---|
| Description | Test the creation and editing of user profiles, including personal information such as name, contact details, and language. |
| Preconditions | User is logged in. |
| Inputs | Users edit their profile information. |
| Expected Output | User profile is updated with the edited information. Changes are saved and reflected in the profile. |

Test Case 3: Course Registration

| Test Case ID: TC_UR_003 | |
|--------------------------------|--|
| Description | Test if users can select courses from a predefined list. |
| Preconditions | User is logged in. |
| Inputs | User selects courses from the list. |
| Expected Output | Users are enrolled in the selected courses. Enrolled courses are listed in the user's profile. |

Test Case 4: Live Search

| Test Case ID: TC_UR_004 | |
|--------------------------------|--|
| Description | Test the live search feature during registration, allowing users to quickly find and select their courses. |
| Preconditions | User is on the registration page. |
| Inputs | User begins typing in the search field. |
| Expected Output | Search results dynamically update as the user types. Users can easily select courses. |

Test Case 5: Language Preference

| Test Case ID: TC_UR_005 | |
|--------------------------------|--|
| Description | Test if users can set their preferred teaching or learning language during registration. |
| Preconditions | User is on the registration page. |

| | |
|------------------------|------------------------------------|
| Inputs | User selects a preferred language. |
| Expected Output | User's language preference is set. |

4.2 Scheduling

Test Case 6: Class Enrollment

| | |
|---------------------------------|--|
| Test Case ID: TC_SCH_001 | |
| Description | Verify that students can view the list of classes they are enrolled in. |
| Preconditions | Student is logged in. |
| Inputs | Students access their enrolled classes. |
| Expected Output | List of enrolled classes is displayed. Classes are accurate and up-to-date. |

Test Case 7: Tutor Selection

| | |
|---------------------------------|--|
| Test Case ID: TC_SCH_002 | |
| Description | Test if students can access information about available tutors for their classes. |
| Preconditions | Student is logged in and has enrolled in at least one class. |
| Inputs | Student selects a class and views available tutors. |
| Expected Output | List of tutors for the selected class is displayed. Tutor information is accurate and includes contact details. |

Test Case 8: Tutor Scheduling

| | |
|---------------------------------|--|
| Test Case ID: TC_SCH_003 | |
| Description | Verify that students can schedule tutoring sessions with available tutors. |
| Preconditions | Student is logged in and has selected a tutor. |
| Inputs | Students schedule a tutoring session. |

| | |
|------------------------|---|
| Expected Output | Tutoring sessions are scheduled and added to the student's calendar. Students and tutors receive confirmation. |
|------------------------|---|

Test Case 9: Tutor Management

| | |
|---------------------------------|---|
| Test Case ID: TC_SCH_004 | |
| Description | Test if tutors can view the list of classes they have chosen to tutor for during registration. |
| Preconditions | Tutor is logged in. |
| Inputs | Tutor accesses their list of tutable classes. |
| Expected Output | List of classes available for tutoring is displayed. Tutors can manage their selected classes. |

Test Case 10: Tutor Requests

| | |
|---------------------------------|--|
| Test Case ID: TC_SCH_005 | |
| Description | Verify that tutors can manage incoming tutoring requests from students, accepting or declining them as needed. |
| Preconditions | Tutor is logged in and has pending requests. |
| Inputs | Tutor views and responds to a tutoring request. |
| Expected Output | Tutor accepts or declines the request, and the student is notified. Request status is updated. |

Test Case 11: Search Filters

| | |
|---------------------------------|--|
| Test Case ID: TC_SCH_006 | |
| Description | Test if users can search for tutors using filters such as course name, course code, preferred learning language, and department. |
| Preconditions | User is logged in. |
| Inputs | User applies search filters. |
| Expected Output | Filtered list of tutors is displayed. Filters are applied accurately. |

4.3 Communication

Test Case 12: In-App Messaging

| Test Case ID: TC_COM_001 | |
|---------------------------------|--|
| Description | Verify that users can communicate with each other via an in-app messaging system. |
| Preconditions | Users are logged in. |
| Inputs | User sends a message to another user. |
| Expected Output | Message is sent and received in real-time. Message content is preserved accurately. |

Test Case 13: Real-Time Chat

| Test Case ID: TC_COM_002 | |
|---------------------------------|---|
| Description | Test whether messaging is real-time, leveraging real-time messaging SDKs to ensure immediate communication. |
| Preconditions | Users are logged in and engaged in a chat. |
| Inputs | Users exchange messages in real-time. |
| Expected Output | Messages are delivered instantly. Chat remains synchronized across devices. |

Test Case 14: Email Integration

| Test Case ID: TC_COM_003 | |
|---------------------------------|--|
| Description | Test the option for users to communicate through email if they prefer email-based communication. |
| Preconditions | Users are logged in. |
| Inputs | User chooses to send a message via email. |
| Expected Output | An email is sent to the recipient's email address. Email content is correct and includes the message. |

4.4 Push Notifications/Alerts

Test Case 15: Appointment Alerts

| Test Case ID: TC_PUSH_001 | |
|----------------------------------|--|
| Description | Verify that users receive push notifications and alerts for appointment updates, including accepted and declined appointments. |
| Preconditions | Users have scheduled appointments. |
| Inputs | User receives an appointment update. |
| Expected Output | Push notification or alert is received promptly. Notification content is accurate and informative. |

Test Case 16: Reminder Alerts

| Test Case ID: TC_PUSH_002 | |
|----------------------------------|---|
| Description | Test whether users receive reminders for scheduled tutoring sessions. |
| Preconditions | Users have upcoming tutoring sessions. |
| Inputs | Users' scheduled tutoring sessions are approaching. |
| Expected Output | Reminder notification is sent before the session. Reminders are received in a timely manner. |

4.5 Student Reviews and Ratings

Test Case 17: Review Submission

| Test Case ID: TC_REV_001 | |
|---------------------------------|--|
| Description | Test if students can submit reviews and ratings after each tutoring session. |
| Preconditions | User has completed a tutoring session. |
| Inputs | Student provides a review and rating for a session. |
| Expected Output | Review and rating are submitted and recorded accurately. Review content is displayed correctly. |

Test Case 18: Review Visibility

| Test Case ID: TC_REV_002 | |
|---------------------------------|---|
| Description | Verify that reviews and ratings are visible to other students considering the same tutor. |
| Preconditions | User is searching for a tutor. |
| Inputs | User views tutor profiles. |
| Expected Output | Reviews and ratings are displayed on the tutor's profile. Reviews are accurate and up-to-date. |

4.6 Event Tracking

Test Case 19: User Engagement Metrics

| Test Case ID: TC_ET_001 | |
|--------------------------------|---|
| Description | Verify that the application integrates a third-party SDK to monitor user engagement, tracking activities such as sign-ups, appointments booked, cancellations, and other relevant interactions. |
| Preconditions | User interacts with the app. |
| Inputs | User performs various actions within the app. |
| Expected Output | The third-party SDK accurately tracks user engagement metrics. Data is collected and updated in real-time. |

Test Case 20: Data Collection

| Test Case ID: TC_ET_002 | |
|--------------------------------|--|
| Description | Test if the system collects data on user behavior within the app. |
| Preconditions | User interacts with the app. |
| Inputs | User performs actions like sign-ups, booking appointments, and cancellations. |
| Expected Output | User behavior data is collected and stored. The data is accurate and includes relevant information. |

4.7 User Engagements and Retention

Test Case 21: Reward System

| Test Case ID: TC_UE_001 | |
|-------------------------|--|
| Description | Verify that the app encourages users to stay engaged and active through a reward system. |
| Preconditions | User performs actions that qualify for rewards. |
| Inputs | User reaches a milestone or completes a reward-earning action. |
| Expected Output | User is rewarded as per the defined reward system. |

5. Testing Risks

1. **Cross-Platform Compatibility:** As the app is developed for multiple platforms (iOS and Android), ensuring consistent functionality and user experience across different devices and operating systems can be challenging. Testing on various devices, screen sizes, and OS versions is crucial to identify and address compatibility issues. Additionally, factors like device-specific features, OS updates, and vendor-specific customizations can further complicate the testing landscape.
2. **Integration with Third-Party Services:** The app relies on several third-party services such as real-time messaging SDKs, push notification services, and ad providers. Testing the integration and reliability of these services, as well as their impact on the app's performance, is essential. Furthermore, ensuring that these services remain consistent and reliable under various network conditions is crucial to prevent unexpected disruptions.
3. **Data Security and Privacy:** Handling user data, including personal information and communication, requires rigorous testing for data security and privacy compliance. Any data breaches or vulnerabilities could lead to legal and reputational risks.
4. **Limited API Access:** The app's reliance on data from Florida Tech's existing API and web hooks poses a risk if the API is not consistently available or if its structure changes. Continuous monitoring and contingency plans are necessary to address potential disruptions. Periodic validation of the API endpoints, response time, and data integrity will be vital in ensuring data consistency.

5. **Server-Side JavaScript Development:** Using JavaScript for server-side automations, especially as the team lacks experience, can introduce risks related to security, scalability, and the reliability of server-side components. Extensive testing and potentially collaborating with experienced backend developers are essential.

6. Testing Resources

6.1 Testing Team

- Sidney Nedd (Test Lead)
- Eleanor Barry (Tester)
- Samaher Damanhori (Tester)

6.2 Hardware Resources

- iOS and Android smartphones and tablets for testing
- Test devices with varying screen sizes and OS versions

6.3 Software Resources

- Mobile operating systems: iOS, Android
- Test management tool (Specify if applicable)
- Test automation tool (Specify if applicable)
- Integrated development environments (IDEs) for mobile app development

6.4 Test Data

- Sample user profiles for testing user registration
- Predefined course lists for course registration
- Mock tutoring schedules for scheduling tests
- Simulated in-app messaging data for communication tests
- Sample push notifications for push notifications/alerts tests
- Test user reviews and ratings for student reviews and ratings tests
- Simulated user engagement data for event tracking tests

6.5 Test Environment

- Test environment configured to simulate real-world usage scenarios

7. Test Execution

- Test execution will follow the test plan and test cases outlined in this document.
- Testers will perform both manual and automated testing, as applicable.
- Test results, including pass/fail status and defects found, will be documented.

8. Reporting

- Test execution reports and defect reports will be generated and shared with the project team.
- Defects will be tracked, and their status will be regularly updated.
- A test summary report will be prepared to provide an overview of testing activities and findings. This report will also provide recommendations for future testing cycles and potential areas of improvement

9. Conclusion

In conclusion, this test plan outlines the testing strategy and approach for the TutorFIT mobile application. It covers various features, including user registration, scheduling, communication, push notifications/alerts, student reviews and ratings, event tracking, and user engagement and retention. The testing process will involve multiple testing types, including functional, usability, performance, security, and compatibility testing.

Upon the successful completion of testing, the application will be evaluated for readiness, and any identified issues or defects will be addressed. The testing team is committed to ensuring the quality and reliability of the TutorFIT app to provide a seamless experience for students and tutors at the Florida Institute of Technology.

Lessons learned during testing will be documented and used to improve future testing processes and practices. Continuous communication and collaboration with stakeholders, including the project team and client, will be essential throughout the testing phase.