

SW500 -Software Engineering

“ITU-SAT”

**(ITU-Student Attendance Tracker)**

Mentor - **Roxanna Rohde** Prepared by:

**Mahima Ganapati (89492)**

**Tatiana Solonets (89659)**

**Yifeng Huang (89980)**

**Shruti Kamati (89832)**

**Sriram Rathinavelu (89717)**

**Ramsha Deshmukh (88587)**

# 

# 

# **Acknowledgments**

We would like to express our gratitude to our esteemed Professor **Roxana Rohde**, for her guidance, inspiration and constructive suggestions that helped us in the preparation of this project. We thank her for giving us the opportunity to explore more in the field of Software Engineering and deal with the techniques used in the real world.

**Content**

**Introduction & Goal**

**What Is ITU-SAT?**

**Need of ITU-SAT**

**Scrum & RAD Team**

**Requirements**

* **Functional Requirement**
* **Non Functional Requirement**

**Project Schedule (GANTT Chart)**

**Software Architecture**

**UML Diagrams**

* **Use case diagram**
* **sequence diagram**

## **Quality and Test Plan**

**Test Cases**

**UI Screens (Andriod/IOS)**

**Conclusion**

**Introduction & Goal:**

Generally a **tracking system** is used for the observing of persons or objects on the move and supplying a timely ordered sequence of respective location data to a model. In the world of mobile based application. Looking at our dependency on our mobile phones,Our goal is to make SAT, a mobile app which automated the attendance of students at ITU. This product will suppress the traditional method of marking attendance with pen and paper, or calling loud roll calls. It simplify and provide enhanced experience of tracking attendance using a cell phone. It will capture students attendance on two criteria- Student’s ID and Student’s current location at the ITU campus. SAT will reduce the hassle for professor and for students of missing the name or mispronouncing it.

**More about Mobile phone services:**

Location-based services or LBS is a term that is derived from the telematics and telecom world. The combination of A-GPS, newer GPS and cellular locating technology is what has enabled the latest “LBS” for handsets and PDAs. Line of sight is not necessarily required for a location fix. This is a significant advantage in certain applications since a GPS signal can still be lost indoors. As such, A-GPS enabled cell phones and PDAs can be located indoors and the handset may be tracked more precisely. This enables non-vehicle centric applications and can bridge the indoor location gap, typically the domain of RFID (radio frequency identification)and RTLS (real time locating systems), with an off the shelf cellular device.

Currently, A-GPS enabled handsets are still highly dependent on the Location-Based Service (LBS) carrier system, so handset device choice and application requirements are still not apparent. Enterprise system integrators need the skills and knowledge to correctly choose the pieces that will fit the application and geography.

**What Is ITU-SAT?**

ITU-SAT is an innovative idea of introducing an automated attendance system at ITU campus. ITU Student Attendance Tracker (SAT) records student attendance based on their voice sample and their GPS location. It makes ITU faculty and student life easy, they can view and manage their accounts independently w.r.t. attendance record at any instance. Since its a mobile application it is compatible with major platforms like IOS and android. This product will suppress the traditional method of marking attendance with pen and paper, or calling loud roll calls. It simplify and provide enhanced experience of tracking attendance using a cell phone. It will capture students attendance on two criteria- Student’s ID and Student’s current location at the ITU campus. SAT will reduce the hassle for professor and for students of missing the name or mispronouncing it.

**Need of ITU-SAT**

1. To simplify the process of taking attendance in class.
2. To reduce the hassle of professors as well as for students for missing the names or mispronunciations.
3. To keep a proper track of students attendance.
4. To automate the process which reduce the chance of errors to great extend.
5. To retrieve information of any student at any instance.

# **ITU-SAT Users**

1. **ITU Students-** They can register themselves and then can give class attendance as per their enrollment in the courses. They can request for an attendance record at any instances.
2. **ITU Professors-** They can register themselves and then add their courses details. They can also request group or individual student attendance record at any instances.

### **ITU-SAT Mobile attendance tracking app benefits:**

* No more double entering your data. Enter it once and it's available any time and any where.
* All site features are also available on mobile devices.
* Interface was designed specifically to be used on mobile devices so it's simple and fast.
* Increase student success rate due to ease in keeping a track of number of classes left and already attended.

### Simplify Attendance Taking

**Scrum & RAD Team:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Name** | **Role** | **Email** |
| 1 | Ramsha Deshmukh | Project Manager |  |
| 2 | Sriram Rathinavelu | Architect |  |
| 3 | Yifeng Huang | Devs (IOS) |  |
| 4 | Tatania | Devs (Android) |  |
| 5 | Shruti | Tester (IOS) |  |
| 6 | Mahima | Tester (Android) |  |

**Requirements**

* **Functional Requirement**

Functional requirements describe what a software system should do i.e.; defines a function of a system and its components. Below are the functional requirements which are identified for the application.

R1: System must allow new user to launch ITU-SAT application.

R2: System must allow user to Login/Register for creating an account based on their roles at ITU-SAT.

R2.1.1: System must allow existing Professor to login and display professor dashboard

R2.1.2: System must allow existing Student to login and display student dashboard

R2.2: System must verify the role of the user (Professor/Student)

R2.2.1: System must register the professor

R2.2.2 : System must register the student

R2.2.3: System must record/register the student’s voice sample while registering the student

R3. System must allow the user to view the dashboard according to their roles

R3.1: Professor Dashboard

R3.2: Student Dashboard

R.3.1.1: Professor Manage Classes [weekday/weekend] on the dashboard

R3.1.1.1: Professor view details

R3.1.1.2: Professor edit class details

R3.1.1.3: Professor delete class

R3.1.1.4: Professor add class

R3.1.1.5: Professor adds weekday class

R3.1.1.6: Professor adds weekend class

R3.1.2: Professor Generate report

R3.1.2.2: Professor receives Excel file on email of attendance (single/group)

R3.1.3: Professor takes Manual Attendance (if req)

R3.2.1: Student Manage Classes [weekday/weekend] on the dashboard

R3.2.1.1:Student Views class list

R3.2.1.2: Student delete drop/delete class

R3.2.1.3: Student Add Class

R3.2.2: System must allow Student to CheckIn on Menu available on their dashboard

to mark attendance.

R3.2.3: System must allow student to view report of the attendance w.r.t the classes they enrolled on their email.

* **Non Functional Requirements:**

**Performance Requirements**

1. Login in time must have a threshold of 30 secs at max.
2. Voice recording must be uploaded in 10 secs at max.
3. Operation loading must be 10 secs at max.

**Alize-Voice recognition**

1. Alize-Voice recognition recognize, compare and sort the voice samples with current voice samples.
2. Alize must consider the threshold accurately.
3. Alize library support backend of the software, by providing the approval or rejection of comparing voice samples.

**GPS coordinate**

1. GPS coordinates should be accurately plotted as per the classrooms assigned for particular courses.
2. Minute variations in the coordinates can be overlooked.

**Safety Requirements**

1. The system must have a threshold for all the operations which are executed in the software application.
2. Error handling for software application while communicating with the Alize voice recognition library and GPS system.
3. The system data storage should be sufficiently large.

**Quality & Test Plan**

The main purpose of the Test Plan Document for ITU SAT is to describe the Objectives, Scope and approach of Software Testing.

**Test Plan Objective**

The main objective of the test plan for ITU SAT system as follows:

* To define the features of the system to be tested.
* To identify and define all the activities necessary to prepare for and to define pass/fail criteria.
* To identify the deliverables of the testing phase.
* To discuss the testing techniques used in ITU SAT system. Prepare test cases based on functional requirements

**Definitions**

The following are some of the terms and definitions used in the ITU SAT system.

* Test: A collection of one or more test cases.
* Test item: A software item that is objective of testing.
* Test plan: A document describing the scope, approach and resources of testing
* Pass and fail criteria: A condition that defines whether test passes or fails
* Test summary report: A document summarizing test results.
* Testing: The process of analyzing a software item to detect the conduct the testing process on the ITU SAT application.

**Graphical user interface testing**

It is the process of [testing](https://en.wikipedia.org/wiki/Software_testing) a product's [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface) .ITU SAT system’s UI is tested for the correct orderly navigation of UI screens and to ensure it meets the specifications.

**Backend Testing**

Used REST API CURL calls to directly tests the products functionality with respect to backend system.It is a kind of Gray box testing

**Functional Testing**

It describes what the system does. It is a type of [black-box testing](https://en.wikipedia.org/wiki/Black-box_testing) that bases its test cases on the specifications of the software component under test. Functions of ITU SAT are tested by feeding them input and examining its output

**Integration Testing**

In Integration testing, individual modules are combined into a single group and tested as a system. The inputs to Integration testing are the modules that are unit tested. This is mainly used to verify functional, performance and reliability requirements. The purpose of integration testing is to find the inconsistencies between the combined software units of ITU SAT system

**Regression Testing**

This is used to test new software bugs or errors in functional areas of the system after changes such as enhancements, patches or configuration changes are performed. The main aim of the regression testing is to ensure that no errors are introduced in the system after the changes and a change in one part of the software doesn’t affect the other. The old test cases are executed again to check the program behavior and previously fixed errors.

The below test plan in IEEE standard provides the details of the test cases that will be executed for testing the application and test plan is done as per the IEEE 829 format.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **REQ\_ ID** | **Test\_ID** | **Test Objective** | **Steps** | **Expected Result** | **Actual Result** | **Status**  Android | **Status**  IOS |
| R1 | T1 | Application Launch | User launches the SAT application | Welcome Screen displayed | Welcome screen displayed | Pass | Pass |
| **REQ\_ ID** | **Test\_ID** | **Test Objective** | **Steps** | **Expected Result** | **Actual Result** | **Status**  Android | **Status**  IOS |
| R2.1.1 | T2.1.1 | Sign -In  (Professor) | Professor Signs-In by providing following details:   * Email:Richard.sun@itu.edu * Password:abc123#(Min: 7 characters) * Clicks on Sign-In | Professor's dashboard displayed | Professor's dashboard displayed | Pass | Pass |
| R2.1.2 | T2.1.2 | Sign-In  (Student) | Student Signs-In by providing following details:   * Email:89492.ganapati@students.itu.edu * Password:abc123#(Min: 7 characters) * Clicks on Sign-In | Student's dashboard displayed | Student's dashboard displayed | Pass | Pass |
| R2.2.1 | T2.2.1 | Registration  (Professor) | New Professor Registers by providing following details:   * First name: Richard * Last name: Sun * Email:Richard.sun@itu.edu * Password: abc1234 * Profile pic(optional) | Professor's dashboard displayed | Professor's dashboard displayed | Pass | Pass |
| R2.2.2 | T2.2.2 | Registration  (Student) | New Student Registers by providing following details:   * First name: Mahima * Last name:Ganapati * Gender:Female * Email:89492.ganapati@students.itu.edu * Student ID :89492 * Click on next | Student voice sample registration screen displayed | Student voice sample registration screen displayed | Pass | Pass |
| **REQ\_ ID** | **Test\_ID** | **Test Objective** | **Steps** | **Expected Result** | **Actual Result** | **Status**  Android | **Status**  IOS |
| R2.2.3 | T2.2.3 | Registration  (Student voice sample) | New student registers his voice sample by following steps   * Hold microphone icon to record. * Speak slow and clear. * Record statement- "My voice is my attendance"or any statement. * Release microphone icon, once done recording. * Click on submit button | Student's dashboard displayed | Student's dashboard displayed | Pass | Pass |
| R3.1 | T3.1 | Professor Dashboard | Professor dashboard consists   * Menu Tab   Menu Tab consists   * Manage classes * Generate reports * Manual attendance | Professor dashboard with Menu and its sub options displayed | Professor’s dashboard with Menu and its sub options displayed | Pass | Pass |
| R3.2 | T3.2 | Student Dashboard | Student dashboard consists   * Menu Tab   Menu Tab consists   * Check-In * Manage classes * View Reports | Student dashboard with Menu and its sub options displayed | Student dashboard with Menu and its sub options displayed | Pass | Pass |
| R3.1.1 | T3.1.1 | Manage class  (Professor) | Professor Manage Class consists lists of all the subjects added by the professor | List of all subjects added by the professor is displayed | List of all subjects added by the professor is displayed | Pass | Pass |
| R3.1.1.1 | T3.1.1.1 | View Class Details  (Professor) | Professor views and edits the already added classes by following steps:   * Swipes the class tab which needs to be viewed/edited to left * selects the edit button * Clicks on edit option to view/ edit the fields | Screen with selected class details is displayed | Screen with selected class details is displayed | Pass | Pass |
| **REQ\_ ID** | **Test\_ID** | **Test Objective** | **Steps** | **Expected Result** | **Actual Result** | **Status**  Android | **Status**  IOS |
| R3.1.1.2 | T3.1.1.2 | Edit Class Details  (Professor) | Professor edits the already added classes by following steps:   * Swipes the class tab which needs to be viewed/edited to left * selects the edit button * Clicks on edit option to edit the fields | Screen with selected class details is displayed | Screen with selected class details is displayed | Pass | Pass |
| R3.1.1.3 | T3.1.1.3 | Delete Class Details  (Professor) | Professor deletes the already added classes by following steps:   * Swipes the class tab which needs to be deleted to left * selects the delete button * Clicks on delete option to delete the class details | The selected class is deleted from the list. | The selected class is deleted from the list. | Pass | Pass |
| R3.1.1.4 | T3.1.1.4 | Add Class Details  (Professor) | Class can be added by following steps:   * Clicks on Menu * Clicks on Manage classes * Clicks on Add * Chooses Weekday class in switch button by swiping the ON/OFF switch button to OFF * Chooses Weekend class in switch button by swiping the ON/OFF switch button to ON | Add weekday or weekend class details screen displayed | Add weekday or weekend class details screen displayed | Pass | Pass |
| **REQ\_ ID** | **Test\_ID** | **Test Objective** | **Steps** | **Expected Result** | **Actual Result** | **Status**  Android | **Status**  IOS |
| R3.1.1.5 | T3.1.1.5 | Add Weekday Class Details  (Professor) | Professor provides following details   * Course Name: Software Engineering * Course code : SW 500 * Course Section: 1 * Class Room : 302 * Duration: 09/14/2015 To: 01/10/2016 * Timings: 18:00 To: 20:45 * Day of Week : Monday * Trimester : Fall 2015 * Clicks on Add | Weekday class is added to the list | Weekday class is added to the list | Pass | Pass |
| R3.1.1.6 | T3.1.1.6 | Add Weekend Class Details  (Professor) | Professor provides following details   * Course Name: Software Engineering * Course code : SW 500 * Course Section: 2 * Class Room : 302 * Add Dates: Chooses Dates on Calendar as : 10/17/2015 , 10/18/2015 , 11/01/2015, 11/02/2015 & 12/10/2015, 12/11/2015 * Timings: 9:00 to 18:00 * Clicks Add | Weekend class is added to the list | Weekend class is added to the list | Pass | Pass |
| R3.1.2 | T3.1.2 | Generate Attendance Report  (Professor) | Professor gets the whole class attendance reports subjectwise by following:   * clicks on menu. * clicks on generate reports. * selects the subject. | Excel sheet with whole class attendance report for the selected subject is sent to Professor’s email | Excel sheet with whole class attendance report for the selected subject is sent to Professor’s email | Fail | Pass |
| R3.1.3 | T3.1.3 | Manual Attendance | Professor clicks on Manual attendance from Menu then fills following details:   * Date: 12/21/2015 * Student Id: 89492 | Manual attendance marked successfully | Manual attendance marked successfully | Fail | Pass |
| **REQ\_ ID** | **Test\_ID** | **Test Objective** | **Steps** | **Expected Result** | **Actual Result** | **Status**  Android | **Status**  IOS |
| R3.2.1 | T3.2.1 | Manage class  (Student) | Student Manage Class consists lists of all the subjects added by the students | List of all subjects added by the student is displayed | List of all subjects added by the students is displayed | Pass | Pass |
| R3.2.1.1 | T3.2.1.1 | View Class Details  (Student) | Student views the already added classes by following steps:   * Swipes the class tab which needs to be viewed to left * selects the view button * Clicks on view option to view the fields | Screen with selected class details is displayed | Screen with selected class details is displayed | Pass | Pass |
| R3.2.1.2 | T3.2.1.2 | Delete Class Details  (Student) | Student deletes the already added classes by following steps:   * Swipes the class tab which needs to be deleted to left * selects the delete button * Clicks on delete option to delete the class details | The selected class is deleted from the list. | The selected class is deleted from the list. | Pass | Pass |
| R3.2.1.3 | T3.2.1.3 | Add Class  (Student) | Students Adds multiple courses by following steps:   * Student enters the course code : 500 * Application auto populates related course name with different sections. * Selects the related course section * Clicks on ADD button | Class added to the list | Class added to the list | Pass | Pass |
| R3.2.2 | T3.2.2 | Check-In  (Student) | Student gives attendance by following steps   * Go to menu * Click on the check in * Holds the microphone icon,then say "My voice is my attendance" * Then release the microphone button | Student checks-In successfully with valid voice recognition and GPS co-ordinates | Student checks-In successfully with valid voice recognition and GPS co-ordinates | Pass | Pass |
| R3.2.3 | T3.2.3 | Generate Attendance Report  (Student) | Student gets individual attendance report subjectwise by following:   * clicks on menu. * clicks on generate reports. * selects the subject. | Excel sheet with individual attendance report for the selected subject is sent | Excel sheet with individual attendance report for the selected subject is sent to student’s email | Fail | Pass |