

Tactical Communication and Simulation Center

(TCSC)

Manual

Prepared By

Hi-Tech Bangla (Bangladesh) Ltd.

Table of Contents

[1 Introduction 4](#_Toc75348237)

[1.1 Audience 4](#_Toc75348238)

[1.2 Prerequisites 4](#_Toc75348239)

[1.3 Installation and Setting Up 5](#_Toc75348240)

[2 Architecture 5](#_Toc75348241)

[2.1 System Architecture 5](#_Toc75348242)

[2.1.1 System Work Flow 5](#_Toc75348243)

[2.1.2 Radio Flow chart 6](#_Toc75348244)

[2.2 Database Architecture 7](#_Toc75348245)

[2.2.1 User Story 7](#_Toc75348246)

[2.2.2 ER Diagram 10](#_Toc75348247)

[2.3 Network Architecture 11](#_Toc75348248)

[3 Scene 12](#_Toc75348249)

[3.1 Common Interface 12](#_Toc75348250)

[3.1.1 Splash Screen 12](#_Toc75348251)

[3.1.2 Login Screen 12](#_Toc75348252)

[3.2 Student Interface 12](#_Toc75348253)

[3.2.1 Operation Mode Selection Screen 12](#_Toc75348254)

[3.2.2 Check List Screen 12](#_Toc75348255)

[3.2.3 Simulation Screen 12](#_Toc75348256)

[3.2.4 Exam Screen 12](#_Toc75348257)

[3.2.5 Setting Screen 12](#_Toc75348258)

[3.2.6 User Profile Screen 12](#_Toc75348259)

[3.3 Admin Interface 12](#_Toc75348260)

[3.4 Instructor Interface 12](#_Toc75348261)

[4 Scripts 12](#_Toc75348262)

[4.1 Game Controller 12](#_Toc75348263)

[4.2 Inventory Controller 12](#_Toc75348264)

[4.3 Radios 12](#_Toc75348265)

[4.3.1 HF 12](#_Toc75348266)

[4.3.2 VHF 12](#_Toc75348267)

[4.3.3 UHF 12](#_Toc75348268)

[4.3.4 Radio Relay 12](#_Toc75348269)

[4.3.5 Microwave 12](#_Toc75348270)

[4.3.6 Networking 12](#_Toc75348271)

[4.3.7 12](#_Toc75348272)

[4.4 Power 12](#_Toc75348273)

[4.4.1 Power Settings 12](#_Toc75348274)

[4.4.2 Power Source 13](#_Toc75348275)

[4.5 Antennas 13](#_Toc75348276)

[4.6 Utils 13](#_Toc75348277)

[4.7 Enums 13](#_Toc75348278)

[4.8 Debug 13](#_Toc75348279)

[5 Materials 13](#_Toc75348280)

[6 Animations 13](#_Toc75348281)

[7 Sounds 13](#_Toc75348282)

[8 Videos 13](#_Toc75348283)

[9 Prefabs 13](#_Toc75348284)

[10 Models 13](#_Toc75348285)

# Introduction

## Audience

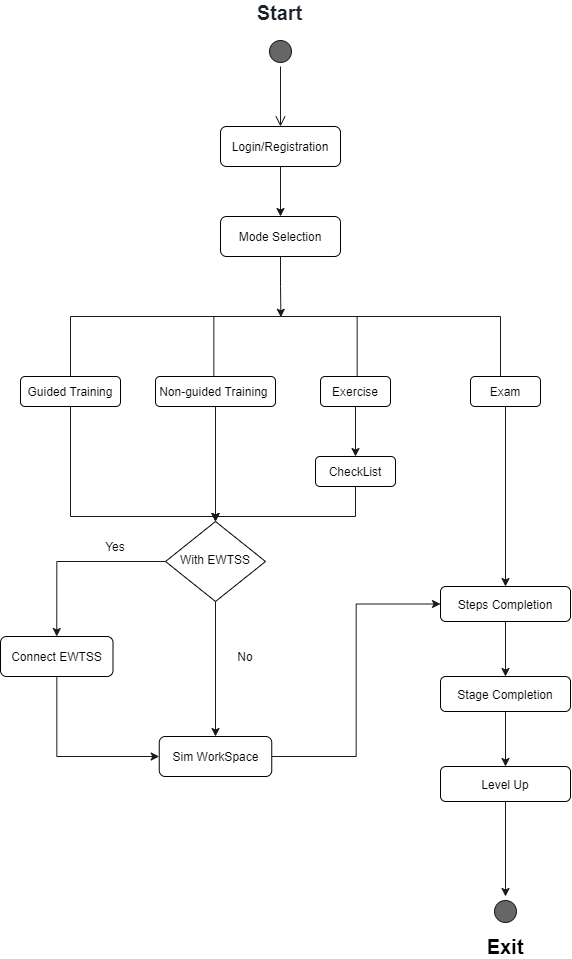
## Prerequisites

## Installation and Setting Up

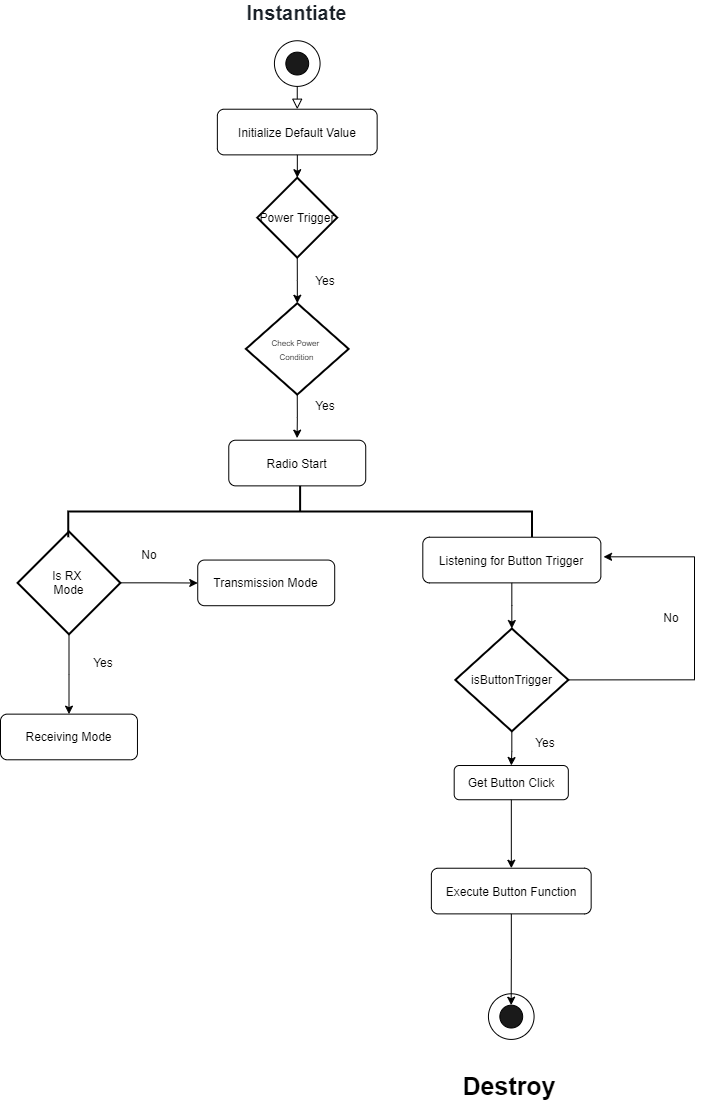
# Architecture

## System Architecture

### System Work Flow



### Radio Flow chart

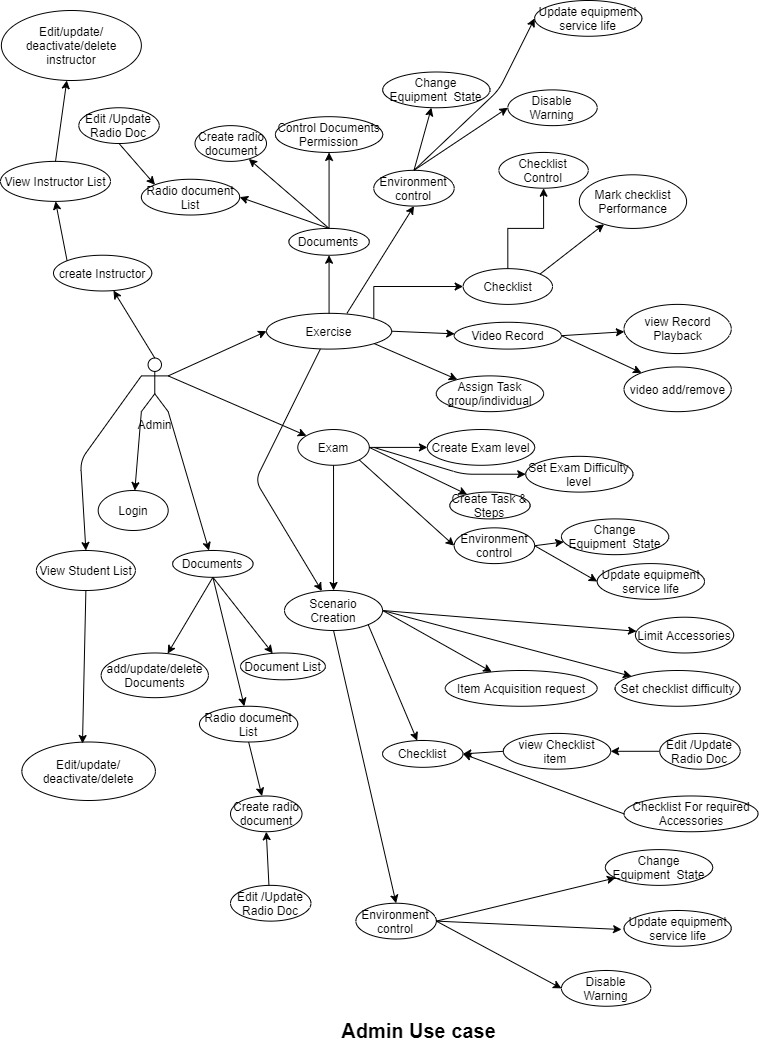


**Fig: 3.1.2 Generic Radio Diagram**

## Database Architecture

### User Story

#### Admin use case Diagram



#### Instructor use case Diagram E:\1. HTBBD\PRDCT\SWFTDev\SMLTR\Militery\TCSC-Army\Alpha\0a_Manual\2_SDD-SoftwareDesignManual\Instructor (1).jpg

#### Student use case DiagramE:\1. HTBBD\PRDCT\SWFTDev\SMLTR\Militery\TCSC-Army\Alpha\0a_Manual\2_SDD-SoftwareDesignManual\Student_Usecase.jpg

### ER Diagram

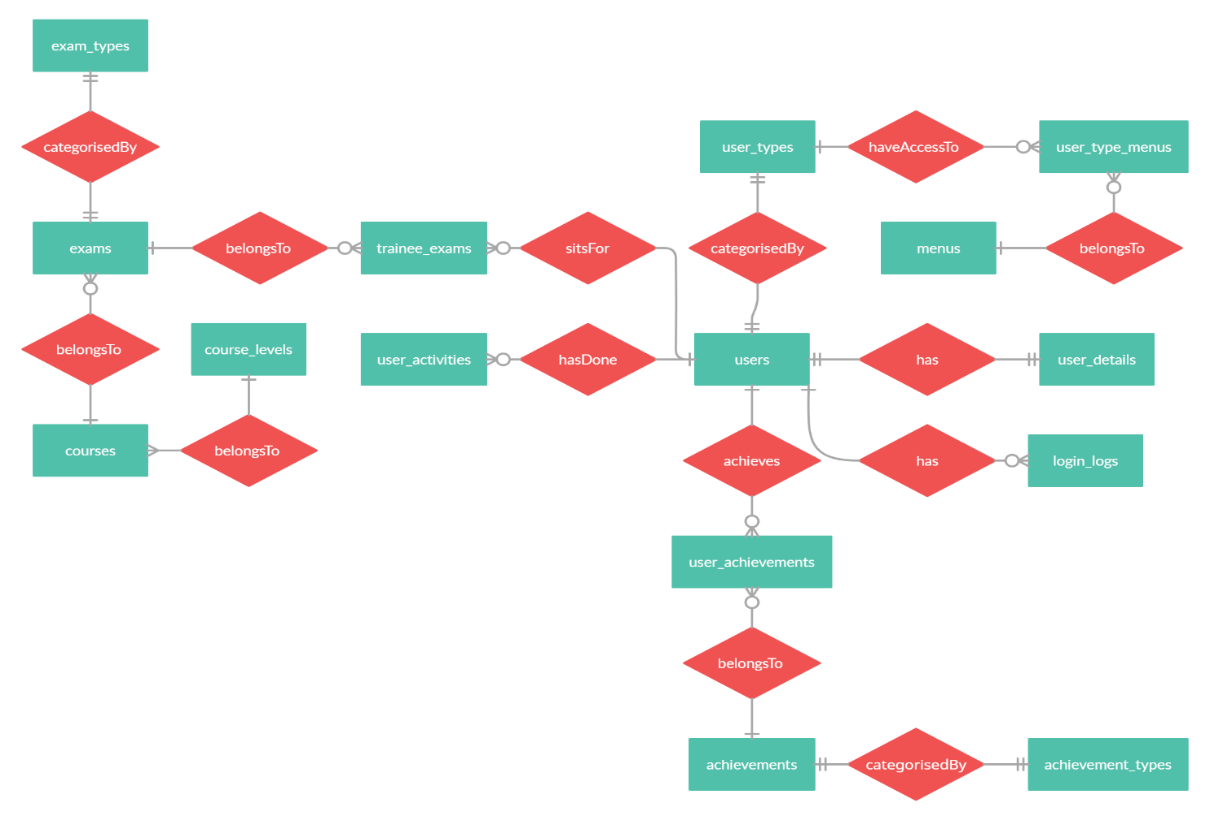


Fig: User ER Diagram

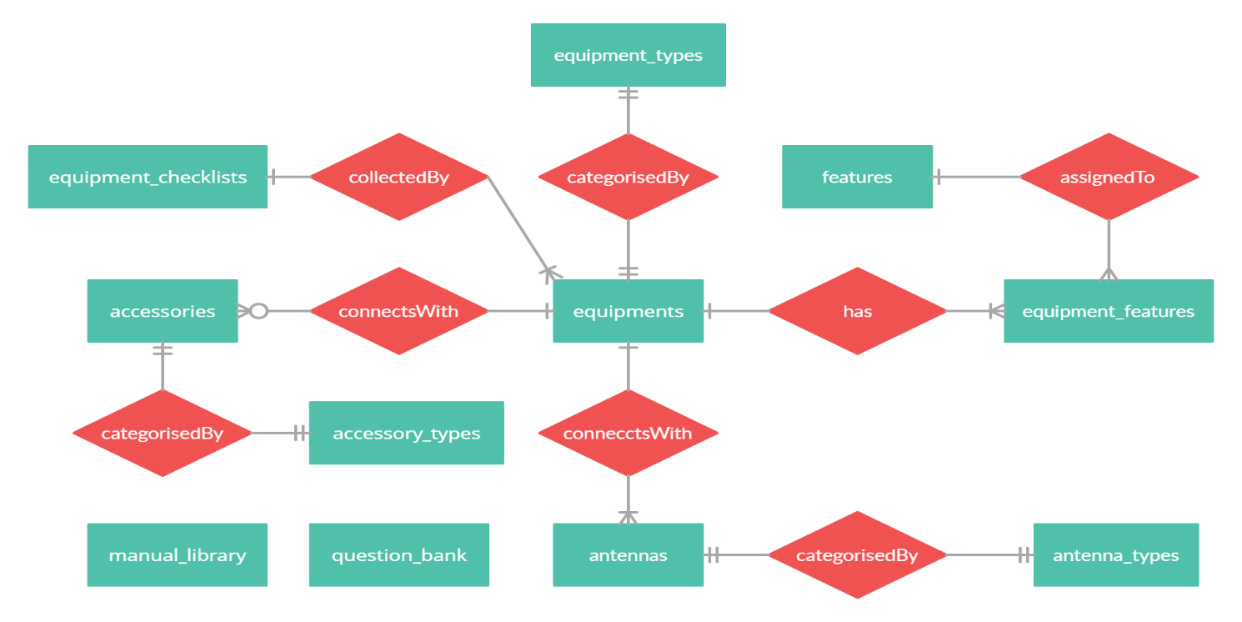
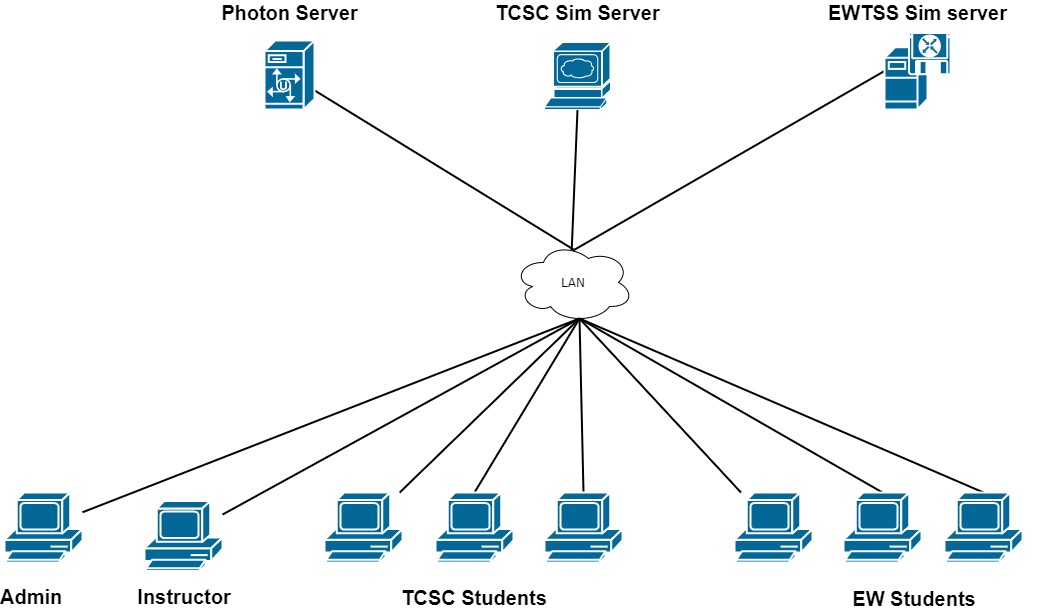


Fig: Equipment ER Diagram

## Network Architecture



# Scene

## Common Interface

### Splash Screen

### Login Screen

## Student Interface

### Operation Mode Selection Screen

### Check List Screen

### Simulation Screen

### Exam Screen

### Setting Screen

### User Profile Screen

## Admin Interface

## Instructor Interface

# Scripts

## Game Controller

## Inventory Controller

## Radios

### HF

### VHF

### UHF

### Radio Relay

### Microwave

### Networking

## Power

### Class

#### Power Settings

Description:

Radio power on/off button execute from this power settings class. Power voltage comes from power source. When radio is switch on power source class execute and give power to the power settings class for start the radio.

Methods:

PowerOnOff (): Function execute when radio power is on/off

ChangePowerMode (): Function execute when switching the power mode. Like power mode will be change based on the Manpack and base station power mode High, Medium, Low. For switching the power mode power output will be changed.

#### Power Source

Description:

Power source class mainly calculate the voltage, watt, AC/DC power Supply, battery health etc. when radio is on power settings class need power source for open radio. This class is dedicated for supply power source.

Update ():

When radio is on, it will always be consuming the power from power source. And decrement the battery label. Update method is executing for consuming power every time.

powerConsumption (): calculate the current supply to the radio. This function executes when radio is on.

powerCharging (): Function execute when radio is charging.

## Antennas

## Utils

## Enums

## UI

### Class

#### LoginUiController

Description:

Login UI Scene Control class. Controlling the Login scene.

Methods:

LoginButtonTrigger (): Execute when login button will press

LoggedIn (): After matching the user Id and password load the login Scene from the function.

#### MainButtonControl

Description:

Menu button execute from Mainbuttoncontrol class.

Methods:

HfArrowControl (): Arrow control execute from this class.

VhfArrowControl (): Arrow control execute from this class.

UhfArrowControl (): Arrow control execute from this class.

RrArrowControl (): Arrow control execute from this class.

MwfArrowControl (): Arrow control execute from this class.

#### MenuControl

Description:

Menu control class execute for enabling menu and mode Scene UI.

Methods:

menuEnable (): Function execute for Enable the Menu

modeEnable (): Function execute for Enable the mode.

#### ModeButtonControl

Description:

Mode Button control can dedicate for selecting the mode and show/hide information.

Methods:

Awake (): When system awake, text and arrow will be enabling

ShowTrainingInfo (): Training information show from this function

ShowExerciseInfo (): ShowExerciseInfo show from this function

ShowExamInfo (): ShowExamInfo show from this function

MainInterface (): MainInterface show from this function

#### CheckListUIControll

## DateTime

### Class

#### DateTimeSettings

Description:

Date time settings class is dedicating for controlling all Radio date Time related settings, Alarm settings and Time Management. This class will be call from anywhere where it will be needed using proper functionality.

Methods:

SetRadioTime (): Function Execute When current radio time need to be changes.

PlayAlerm (): Function Execute when current Time match the alarm time.

## Audio

### Class

#### AudioProfile

Description:

Radio Audio controlling class dedicated for all audio and audio related methods. This class is important for managing the audio controlling profile. Use this class with proper methods.

Methods:

MuteAudioSelection (): Function Execute for Radio Mute or unmute

MuteSelectiveCallSelection (): Function executes when Selcall signal strength comes. Mute and unmute for noise related issues.

MuteSslSelection (): Function executes when signal strength comes and need to mute and unmute.

## Call

### Class

#### CallSettings

Descriptions:

Radio Call Settings class Dedicated for call settings. This class is executing directly from radio main class. When call button is press, radio need callsettings class for execute the call. Call settings class also use the addressbook class to get the Contact Unique Id or set the unique id for make the call. After all settings done, press the call button, call switch case will execute.

Methods:

setSelCallId (): Function execute for set the selcallId, and execute call switch case for make calls.

setTeleCallId (): Function execute for set the setTeleCallId, and execute call switch case for make calls.

setHangupCallId (): Function execute for set the setHangupCallId, and execute call switch case for make calls.

setPageCallId (): Function execute for set the setPageCallId, and execute call switch case for make calls. Page call also support massaging system for sending the text message.

setGpsRequestCallId (): Function executes for set the setGpsRequestCallId, and execute call switch case for make calls.

setStatusRequestCallId (): Function execute for set the setStatusRequestCallId, and execute call switch case for make calls.

setSecureCallId (): Function executes for set the setSecureCallId, and execute call switch case for make calls.

setBeaconCallId (): Function executes for set the setBeaconCallId, and execute call switch case for make calls.

# Materials

# Animations

# Sounds

# Videos

# Prefabs

# Models