Automated Wireless Asset Tracking for Underground Mines

Midterm Report

For Tracking & Monitoring Software

Version 1.0

Table of Contents

[Literature Review 5](#_Toc404262452)

[Modern Software Requirements Specification 6](#_Toc404262453)

[1. Introduction 6](#_Toc404262454)

[1.0.1 Purpose 6](#_Toc404262455)

[1.0.2 Scope 6](#_Toc404262456)

[1.0.3 Definitions, Acronyms and Abbreviations 6](#_Toc404262457)

[1.0.4 References 7](#_Toc404262458)

[1.0.5 Overview 7](#_Toc404262459)

[2. Overall Description 8](#_Toc404262460)

[2.0.1 Use-Case Model Survey 8](#_Toc404262461)

[2.0.2 Assumptions and Dependencies 9](#_Toc404262462)

[3. Requirements 10](#_Toc404262463)

[3.0.1 Use-Case Specifications 10](#_Toc404262464)

[3.0.2 Functionality 10](#_Toc404262465)

[3.0.3 Usability 10](#_Toc404262466)

[3.0.4 Reliability 10](#_Toc404262467)

[3.0.5 Performance 10](#_Toc404262468)

[3.0.6 Supportability 10](#_Toc404262469)

[3.0.7 Design Constraints 11](#_Toc404262470)

[3.0.8 Online User Documentation and Help System Requirements 11](#_Toc404262471)

[3.0.9 Purchased Components 11](#_Toc404262472)

[3.0.10 Interfaces 11](#_Toc404262473)

[3.0.11 Licensing Requirements 12](#_Toc404262474)

[3.0.12 Legal, Copyright and Other Notices 12](#_Toc404262475)

[3.0.13 Applicable Standards 13](#_Toc404262476)

[Software Design Specification 14](#_Toc404262477)

[4. Introduction 14](#_Toc404262478)

[4.0.1 Purpose of this document 14](#_Toc404262479)

[4.0.2 Scope of the development project 14](#_Toc404262480)

[4.0.3 Definitions, acronyms, and abbreviations 14](#_Toc404262481)

[4.0.4 References 14](#_Toc404262482)

[4.0.5 Overview of document 14](#_Toc404262483)

[5.0 Logical Architecture 15](#_Toc404262485)

[5.0.1 Overview 15](#_Toc404262486)

[5.0.2 Master Module 15](#_Toc404262487)

[5.0.3 Tracking Module 15](#_Toc404262488)

[5.0.4 Messaging Module 15](#_Toc404262489)

[5.0.5 Attendance Module 15](#_Toc404262490)

[5.0.6 Reports Module 15](#_Toc404262491)

[6. Detailed Description of Components 16](#_Toc404262492)

[6.0.1 Master Module Class Diagram 16](#_Toc404262493)

[6.0.2 Tracking Module Class Diagram 20](#_Toc404262494)

[6.0.3 Messaging Module Class Diagram 25](#_Toc404262495)

[6.0.4 Attendance Module Class Diagram 27](#_Toc404262496)

[6.0.5 Reports Module Class Diagram 30](#_Toc404262497)

[6.0.6 Sequence Diagrams 32](#_Toc404262498)

[6.0.7 Relational Database Schema 41](#_Toc404262499)

[7. Design Rationale 41](#_Toc404262500)

[8. Review Forms 42](#_Toc404262501)

[Walkthrough Report 43](#_Toc404262502)

[Software Testing Plan 44](#_Toc404262503)

[8.0 Introduction 44](#_Toc404262504)

[8.1 Relationship to other documents 44](#_Toc404262505)

[8.2 System overview 44](#_Toc404262506)

[8.3 Features to be tested/not to be tested 44](#_Toc404262507)

[8.4 Pass/Fail criteria 44](#_Toc404262508)

[8.5 Approach 44](#_Toc404262509)

[8.6 Suspension and resumption 44](#_Toc404262510)

[8.7 Testing materials (hardware/software requirements) 44](#_Toc404262511)

[8.8 Test cases 44](#_Toc404262512)

[8.9 Testing schedule 44](#_Toc404262513)

[Gantt Chart 45](#_Toc404262514)

[References 46](#_Toc404262515)

[Use Case Specification Documents 47](#_Toc404262516)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 20/11/14 | 1.0 | Added sequence diagrams to section 6.0.6 | Philip Kurowski |
| 14/11/14 | 1.0 | Completed SDS Introduction  Added class diagrams to section 6.0.1 – 6.0.5 | Philip Kurowski |
| 12/11/14 | 1.0 | Completed applicable standards.  Completed interfaces. | Philip Kurowski |
| 11/11/14 | 1.0 | Completed section 1  Added UCD to section 2  Completed section 3 (minus interfaces, applicable standards, and functionality) | Philip Kurowski |
| 10/10/14 | 1.0 | Filled out SRS introduction. | Philip Kurowski |
| 29/09/14 | 1.0 | Filled out fields. | Philip Kurowski |
|  |  |  |  |
|  |  |  |  |

# Literature Review

# Modern Software Requirements Specification

## Introduction

### Purpose

This software requirements specification describes the external behavior of the tracking and monitoring software system. It also describes nonfunctional requirements, design constraints, and the other factors integral in the description of the tracking and monitoring software.

### Scope

This SRS applies to the Tracking and Monitoring Software used in a ZigBee-based automated wireless asset tracking system for underground mines. The software is associated with the use case diagram found in [**section 2.0.1.4**](#_Diagrams_of_the).

The completed application will let users:

* Track positions of miners and mining vehicles.
* Project miner and mining vehicle paths on a map.
* Specify ZigBee networks nodes including routers, end devices, and sensors.
* Generate attendance reports.
* Generate reports based on network node information, miner information, and vehicle information.
* Communicate via text with miners.

The completed application will not:

* Allow communicate via speech
* Automate communications to end devices (done in coordinator firmware)

### Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Coordinator** | A device that maintains all network knowledge. There is only one in the ZigBee network. |
| **End Device** | A device in end device mode. It can only communicate with coordinators. |
| **FFD** | Full-function device, can be implemented as a coordinator or an end device. |
| **Git / GitHub** | Source control and online hub that will host versions of the described software codebase. |
| **RFD** | Reduced-function device, can only be implemented as an end device. |
| **Router** | In this scope, a router is any FFD that connects to the coordinator. Routers are stationary and provides connectivity for the coordinator and end devices. |
| **TMS** | Acronym for Tracking & Monitoring Software, the software described in this SRS. |
| **Visual Studio** | Integrated development environment for Windows applications. Includes C# integration. |
| **ZigBee** | A wireless packet transfer specification built on IEEE 802.15.4 that uses a single coordinator, multiple routers and multiple end devices to build a low rate control / sensor network. |

### References

The referenced documents are use case specification documents and can be found in the [Use Case Documents](#_Use_Case_Description) section of the report.

|  |  |  |
| --- | --- | --- |
| **Document No.** | **Document Title** | **Date** |
| 1 | AssignShift.docx | 10/05/14 |
| 2 | AssignTag.docx | 10/05/14 |
| 3 | BroadcastText.docx | 10/10/14 |
| 4 | CreateAlert.docx | 10/05/14 |
| 5 | CreateAttendanceReport.docx | 10/17/14 |
| 6 | CreateMember.docx | 10/14/14 |
| 7 | CreateMessageDetailsReport.docx | 10/20/14 |
| 8 | CreateMinerReport.docx | 10/20/14 |
| 9 | CreateMinerBlockReport.docx | 10/20/14 |
| 10 | CreateMinerPositionReport .docx | 10/20/14 |
| 11 | CreateRouterReport .docx | 10/20/14 |
| 12 | CreateSensor.docx | 10/11/14 |
| 13 | CreateTag.docx | 10/05/14 |
| 14 | CreateTagReport.docx | 10/20/14 |
| 15 | CreateUser.docx | 10/20/14 |
| 16 | CreateVehicleOperationReport.docx | 10/15/14 |
| 17 | DailyReport.docx | 10/18/14 |
| 18 | LoadMap.docx | 10/09/14 |
| 19 | LogIn.docx | 10/07/14 |
| 20 | MonthlyReport.docx | 10/18/14 |
| 21 | ReceiveMessage.docx | 10/16/14 |
| 22 | ViewMemberPosition.docx | 10/07/14 |
| 23 | ViewPathChart.docx | 10/05/14 |
| 24 | YearlyReport.docx | 10/18/14 |

### Overview

The rest of the SRS examines the specifications of the Tracking and Monitoring Software. The second section of the SRS describes the use case model and core functionality from a high level point of view. The third section of the SRS will go over the specific program requirements of the Tracking and Monitoring Software, including detailed descriptions of functional, and non-functional requirements along with constraints, licensing, and legal notices.

## Overall Description

### Use-Case Model Survey

#### Introduction

The use-case model provides an overview of the tracking and monitoring software’s core functionalities from a system perspective. The separation of user access levels is shown as are all features of the software that are both initiated by network events and initiated by the user. Use-cases are divided into 5 modules: Master, Tracking, Messaging, Attendance, and Reports.

The use-case model contains the four system actors including a user, an administrator, the network coordinator, and the TMS server itself. Exceptional cases are shown by extends stereotypes and include cases of different attendance reports.

#### Survey Description <NOT DONE>

#### Use-Case Model Hierarchy

##### Tracking & Monitoring Software

The package contains all system modules including those used for administrative tasks, receiving & displaying tracking data, and generating reports. The software functions as a Windows application that automatically tracks ZigBee network devices and lets a user perform all-purpose monitoring of miners and vehicles in an underground mine.

See section 2.1.2 for a list of all package use cases and descriptions.

The system actors are:

* **User**:
* **Admin**:
* **Coordinator**:

#### Diagram of the Use-Case Model



Figure 1: The TMS use case model

### Assumptions and Dependencies

#### Coordinator is up-to-date

The FFD acting as the coordinator is up to date with the latest firmware that can communicate with the TMS.

#### Server can connect

The server running the TMS has the appropriate ports to communicate with the network.

## Requirements

### Use-Case Specifications

All use cases and their corresponding description documents can be found in the [Use Case Specification Documents](#_Use_Case_Specification) section of the report.

### Functionality

#### Display mine plan and structure

The image containing the mining site plan will be displayed along with all routers set in the site.

#### Display Path Chart for Miners

A miner’s path can be drawn on the mining site plan. The path is the visited route of routers that the miner has visited.

#### Display Miner Position

A miner’s position can be highlighted on the map

#### Generate Attendance Reports

Available attendance reports include daily, monthly, and yearly reports for miner attendance and general operation time reports for vehicles.

#### Generate Network Node Reports

Available reports include miners, miner’s position, miners blocked, router, tag, and sent / received message reports.

#### Add / Edit Network Nodes

Routers, sensors, and end devices can be added and edited from the application.

### Usability

#### Training Time

Under an hour to be able to add and edit components, and monitor data.

#### User Interface & Usability Design Standards

Spacing & positioning, size, grouping, and intuitiveness will all be applied in forms according to this standard: <http://msdn.microsoft.com/en-us/library/aa468595.aspx#humanux_topic2>

### Reliability

##### *Availability*

TMS server should be up 99.99% of the time.

### Performance

#### Throughput

At least 10 updates / second should be received from the coordinator to allow real time tracking.

### Supportability

#### MSDN C# Coding Conventions

Microsoft’s C# programming guide as shown as outlined:

<http://msdn.microsoft.com/en-us/library/ff926074.aspx>

### Design Constraints

#### ZigBee Network

The physical system will be a network of a coordinator, several routers and many end devices as described in ZigBee specifications.

#### C# Language

All application layer code will be written in C# as a Windows Forms application.

#### Microsoft Visual Studio 2013

The application source code will be a visual studio solution.

### Online User Documentation and Help System Requirements

N/A

### Purchased Components

#### Full Function Devices (FFDs)

Multiple FFDs will be programmed and used for testing as a coordinator and routers to simulate an underground mine.

#### Reduced Function Devices (RFDs)

Multiple RFDs will be used to test end device functionality.

### Interfaces

#### User Interfaces

##### Master / Tracking Interface

* This will be the main view that is open at all times of operation. From this view, we can access all other views via a menu bar on top of the screen.
* The tracking module will be contained on the right side of the screen as a map image above a legend. The legend shows the images that represent routers, blocked routers, and active miners. Clicking on routers or miners in the map will display a small box that shows information about said router or miner.

##### Messaging Interface

* This interface will appear on the master interface at the top right.
* It will contain a button to send a text and a button to open the list of received texts.
* The sending text window will contain a field for a simple text message and a list of miners to send to.
* The received texts window will contain a list ordered by date received that will show the message text when clicked.

##### Reports Interface

* A simple window containing a drop down of the report types on the left. Selecting a report type will reveal the input method to input the valid parameters for the report.
* When all fields are filled out, a generate report button will be enabled.

##### Attendance Interface

* A simple window containing a list of miner names and a checkbox group of the different attendance report types (Daily, Monthly, Yearly).
* Depending on the report type, the user will be presented with a date time picker, a month picker, or a year picker for daily, monthly, and yearly reports respectively.
* There will be two tabs: Miner, as described above, and Vehicle, which will allow selecting a vehicle and generating an operation time report.

#### Hardware Interfaces

N/A

#### Software Interfaces

##### Coordinator firmware

The TMS will communicate directly with the network coordinator using a custom protocol to exchange network topology information.

#### Communications Interfaces

##### RS-232

The port at which the network coordinator will connect to the server hosting the tracking and monitoring software.

### Licensing Requirements

* Philip Kurowski hereby grants Mircom Group a worldwide, perpetual, non-exclusive, non-transferable license to all software for Fire-Alert’s use in connection with the establishment, use, maintenance and modification of the system implemented by Philip Kurowski. Software shall mean executable object code of software programs and the patches, scripts, modifications, enhancements, designs, concepts or other materials that constitute the software programs necessary for the proper function and operation of the system as delivered by Philip Kurowski and accepted by the Customer.
* Any and all licenses, product warranties or service contracts provided by third parties in connection with any software, hardware or other software or services provided in the system shall be delivered to Mircom Group for the sole benefit of Fire-Alert.
* Mircom Group may supply to Philip Kurowski or allow Philip Kurowski to use certain proprietary information, including service marks, logos, graphics, software, documents and business information and plans that have been authored or pre-owned by Mircom Group. All such intellectual property shall remain the exclusive property of Mircom Group and shall not be used by Philip Kurowski for any purposes other than those associated with delivery of the system.

### Legal, Copyright and Other Notices

* Customer may make copies of the software for archival purposes and as required for modifications to the system. All copies and distribution of the software shall remain within the direct control of Customer and its representatives.
* Customer may make modifications to the source code version of the software, if and only if the results of all such modifications are applied solely to the system. In no way does this Software License confer any right in Customer to license, sublicense, sell, or otherwise authorize the use of the software, whether in executable form, source code or otherwise, by any third parties, except in connection with the use of the system as part of Customer’s business.
* All express or implied warranties relating to the software shall be deemed null and void in case of any modification to the software made by any party other than Philip Kurowski.
* Philip Kurowski represents and warrants to Mircom Group that:
* it has all necessary rights and authority to execute and deliver this Software License and perform its obligations hereunder and to grant the rights granted under this Software License to Customer;
* the goods and services provided by contractor under this Software License, including the software and all intellectual property provided hereunder, are original to Philip Kurowski or its subcontractors or partners; and
* the software, as delivered as part of the system, will not infringe or otherwise violate the rights of any third party, or violate any applicable law, rule or regulation.
* Philip Kurowski further represents and warrants that, throughout the System Warranty Period, the executable object code of software and the system will perform substantially in accordance with the System Specifications and Agreement. If the software fails to perform as specified and accepted all remedies are pursuant to the policies set forth in the Specification and in the Agreement. **No warranty of any type or nature is provided for the source code version of the software which is delivered as is.**
* Except as expressly stated in this Agreement, there are no warranties, express or implied, including, but not limited to, the implied warranties of fitness for a particular purpose, of merchantability, or warranty of no infringement of third party intellectual property rights.

### Applicable Standards

#### C# Coding Standards

All code will adhere to Microsoft’s outlines C# coding conventions:

<http://msdn.microsoft.com/en-us/library/ff926074.aspx>

#### Windows Forms Usability Standards

All UI and windows forms will take into account Microsoft’s UI and Usability standards:

<http://msdn.microsoft.com/en-us/library/aa468595.aspx#humanux_topic2>

# Software Design Specification

## Introduction

### Purpose of this document

The main objectives of the SDS are to provide both a high level description of the system architecture and enough low level details to allow a developer to fully implement the system. The role of each module will be described to provide a high level description of the system and the connectivity between the modules. Each module will then be described by a class diagram and a detailed overview will be presented for each module’s class’ attributes and operations to describe implementation.

### Scope of the development project

The scope defined in [section 1.0.2](#_Scope) applies to the SDS as well.

### Definitions, acronyms, and abbreviations

All definitions found in [section 1.0.3](#_Definitions,_Acronyms_and) apply to the SDS as well.

### References

The SRS associated with this design can be found [prepended to this document](#_Modern_Software_Requirements). Use case specifications are located at the [end of the report](#_Use_Case_Description).

### Overview of document

The rest of the SDS examines the design of the Tracking and Monitoring Software. The second section of the SDS describes the high level design of the TMS’ modules and is followed by a detailed look into the classes associated with each module. The final sections of the design document contains the reasoning for various choices related to architecture and class structure that were made throughout the system design phase.



## Logical Architecture

### Overview

The Tracking & Monitoring Software’s logical architecture is divided into five modules: a Master module for general purpose management tasks, a Tracking module for tasks dealing with observing the mining site network data, a Messaging module for text communication with end devices in the network, and Report and Attendance modules for generating reports based on network data and member attendance. See [section 2.0.1.](#_Diagram_of_the)4 for overall use case diagram.

### Master Module

General management tasks, this module accesses the system database to insert, update, and delete data related to members, network nodes, and users.

### Tracking Module

Primary data visualization, provides an interface for viewing and receiving network data received from the coordinator.

### Messaging Module

Contains data and functions that allow the user to communicate with miners via text messaging to end devices.

### Attendance Module

Provides an interface for the user to generate reports on individual miner attendance.

### Reports Module

Provides an interface for the user to generate reports on all miners, routers, and tags. Manipulates and organized data needed for the reports into an easily readable format.

## Detailed Description of Components

### Master Module Class Diagram



Figure 2: Master module class diagram

#### MasterController

The controller for the master module. This performs all the tasks in the master module of the use case diagram and serves as a façade to access all other modules.

##### MasterController Attributes

N/A

##### MasterController Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| assignShift(miner : Miner, shift : Shift) |  |
| createMember() |  |
| createSensor() |  |
| createUser() |  |
| createTag(tagId : String) |  |
| loadMap() : void |  |
| logIn(username : string, pass : string) |  |
| updateMember() |  |
| updateSensor() |  |
| updateUser() |  |

##### MasterController Design Specification/Constraints

N/A

##### MasterController States and Transitions

N/A

#### LoginForm

The view class providing an interface for the user to log in.

#### LoginForm Attributes

N/A

##### LoginForm Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| OnLoginPress() | Delegate for when the user presses a login button in the view. |

##### LoginForm Design Specification/Constraints

N/A

##### LoginForm States and Transitions

N/A

#### MainForm

The controller for the master module. This performs all the tasks in the master module of the use case diagram and serves as a façade to access all other modules.

##### MainForm Attributes

N/A

##### MainForm Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| OnAttendancePress() |  |
| OnLoadMapPressed() |  |
| OnMessagingAlertPress() |  |
| OnMessagingBroadcastPress() |  |
| OnReportsPress() |  |
| OnTrackingMapClick() |  |

##### MainForm Design Specification/Constraints

N/A

##### MainForm States and Transitions

N/A

#### MemberForm

The controller for the master module. This performs all the tasks in the master module of the use case diagram and serves as a façade to access all other modules.

##### MemberForm Attributes

N/A

##### MemberForm Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| OnAssignShiftPress() |  |
| OnAssignTagPress() |  |

##### MemberForm Design Specification/Constraints

N/A

##### MemberForm States and Transitions

N/A

#### ShiftForm

The controller for the master module. This performs all the tasks in the master module of the use case diagram and serves as a façade to access all other modules.

##### ShiftForm Attributes

N/A

##### ShiftForm Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| OnAssignPress() |  |

##### ShiftForm Design Specification/Constraints

N/A

##### ShiftForm States and Transitions

N/A

#### TagForm

The controller for the master module. This performs all the tasks in the master module of the use case diagram and serves as a façade to access all other modules.

##### TagForm Attributes

N/A

##### TagForm Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| OnUpdateButtonPress() |  |

##### TagForm Design Specification/Constraints

N/A

##### TagForm States and Transitions

N/A

#### UserForm

The controller for the master module. This performs all the tasks in the master module of the use case diagram and serves as a façade to access all other modules.

##### UserForm Attributes

N/A

##### UserForm Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| OnUpdateButtonPress() |  |

##### UserForm Design Specification/Constraints

N/A

##### UserForm States and Transitions

N/A

### Tracking Module Class Diagram



Figure 3: Tracking module class diagram

#### TrackingController

<EachClass.Documentation>

##### TrackingController Attributes

N/A

##### TrackingController Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| <Operation.Name> | <Operation.Documentation> |

##### TrackingController Design Specification/Constraints

N/A

##### TrackingController States and Transitions

N/A

#### Map

<EachClass.Documentation>

##### Map Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| <Attribute.Name> | <Attribute.Type> | <Attribute.Documentation> |

##### Map Operations

N/A

##### Map Design Specification/Constraints

* Image constrained to 800 x 700 pixels

##### Map States and Transitions

N/A

#### Member

<EachClass.Documentation>

##### Member Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| memberNo | String |  |
| fName | String |  |
| mName | String |  |
| lName | String |  |
| Address | String |  |
| Province | String |  |
| City | String |  |
| pinNo | Int |  |
| phoneNo | String |  |
| mobileNo | String |  |
| returnDate | Date |  |
| isVehicle | Boolean |  |

##### Member Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| appendRouter(router : Router) |  |
| getRouterPath() |  |

##### Member Design Specification/Constraints

N/A

##### Member States and Transitions

N/A

#### Router

<EachClass.Documentation>

##### Router Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| id | String |  |
| address | String |  |
| location | String |  |
| x | int |  |
| y | int |  |
| isBlocked | Boolean |  |

##### Router Operations

N/A

##### Router Design Specification/Constraints

N/A

##### Router States and Transitions

N/A

#### RouterInfoForm

<EachClass.Documentation>

##### RouterInfoForm Attributes

N/A

##### RouterInfoForm Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| setPosition(x : int, y : int) |  |
| setRouter(router : Router) |  |

##### RouterInfoForm Design Specification/Constraints

N/A

##### RouterInfoForm States and Transitions

N/A

#### Sensor

<EachClass.Documentation>

##### Sensor Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| Id | String |  |

##### Sensor Operations

N/A

##### Sensor Design Specification/Constraints

N/A

##### Sensor States and Transitions

N/A

#### Shift

<EachClass.Documentation>

##### Shift Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| start | DateTime |  |
| end | DateTime |  |

##### Shift Operations

N/A

##### Shift Design Specification/Constraints

N/A

##### Shift States and Transitions

N/A

#### Site

<EachClass.Documentation>

##### Site Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| name | String |  |

##### Site Operations

N/A

##### Site Design Specification/Constraints

N/A

##### Site States and Transitions

N/A

#### Tag

<EachClass.Documentation>

##### Tag Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| id | String |  |

##### Tag Operations

N/A

##### Tag Design Specification/Constraints

N/A

##### Tag States and Transitions

N/A

#### TrackingPanel

<EachClass.Documentation>

##### TrackingPanel Attributes

N/A

##### TrackingPanel Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| drawPath(points : Point[]) |  |
| selectMember() |  |

##### TrackingPanel Design Specification/Constraints

N/A

##### TrackingPanel States and Transitions

N/A

### Messaging Module Class Diagram



Figure 4: Messaging module class diagram

#### MessagingController

<EachClass.Documentation>

##### MessagingController Attributes

N/A

##### MessagingController Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| getAlertDialog() |  |
| getBroadcastDialog() |  |
| broadcastMessage(content : String) |  |
| createAlert(content : String) |  |

##### MessagingController Design Specification/Constraints

N/A

##### MessagingController States and Transitions

N/A

#### CustomAlert

<EachClass.Documentation>

##### CustomAlert Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| content | String |  |

##### CustomAlert Operations

N/A

##### CustomAlert Design Specification/Constraints

N/A

##### CustomAlert States and Transitions

N/A

#### EDBroadcast

<EachClass.Documentation>

##### EDBroadcast Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| content | String |  |
| time | DateTime |  |

##### EDBroadcast Operations

N/A

##### EDBroadcast Design Specification/Constraints

N/A

##### EDBroadcast States and Transitions

N/A

#### User

<EachClass.Documentation>

#### User Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| username | String |  |
| password | String |  |
| fName | String |  |
| lName | String |  |

##### User Operations

N/A

##### User Design Specification/Constraints

N/A

##### User States and Transitions

N/A

### Attendance Module Class Diagram



Figure 5: Attendance module class diagram

#### AttendanceController

The main controller for the attendance module. This generates the various attendance range reports for selected members.

##### AttendanceController Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| <Attribute.Name> | <Attribute.Type> | <Attribute.Documentation> |

##### AttendanceController Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| generateDaily(start : Date, end : Date) |  |
| generateMonthly(start : Date, end : Date) |  |
| generateYearly(start : Date, end : Date) |  |

##### AttendanceController Design Specification/Constraints

N/A

##### AttendanceController States and Transitions

N/A

#### AttendanceView

The main controller for the attendance module. This generates the various attendance range reports for selected members.

##### AttendanceView Attributes

N/A

##### AttendanceView Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| OnRadioButton() |  |
| setDailyInput() |  |
| setMonthlyInput() |  |
| setYearlyInput() |  |

##### AttendanceView Design Specification/Constraints

N/A

##### AttendanceView States and Transitions

N/A

#### ExportReportView

The main controller for the attendance module. This generates the various attendance range reports for selected members.

##### ExportReportView Attributes

N/A

##### ExportReportView Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| showReport(reportPath : String) |  |

##### ExportReportView Design Specification/Constraints

N/A

##### ExportReportView States and Transitions

N/A

### Reports Module Class Diagram



Figure 6: Reports module class diagram

#### ReportsController

<EachClass.Documentation>

##### ReportsController Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| reportType | Enumerator |  |

##### ReportsController Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| generateMessageDetailsReport() |  |
| generateMinerReport() |  |
| GenerateMinerBlockReport() |  |
| generateMinerPositionReport() |  |
| generateRouterReport() |  |
| generateTagReport() |  |
| generateReport() |  |
| setReportType(reportType : Enumerator) |  |

##### ReportsController Design Specification/Constraints

N/A

##### ReportsController States and Transitions

N/A

#### ReportsView

<EachClass.Documentation>

##### ReportsView Attributes

N/A

##### ReportsView Operations

|  |  |
| --- | --- |
| **Name** | **Description** |
| OnListItemSelected() |  |
| setMessageDetailsInput() |  |
| setMinerInput() |  |
| setMinerBlockInput() |  |
| setMinerPositionInput() |  |
| setRouterInput() |  |
| setTagInput() |  |

##### ReportsView Design Specification/Constraints

N/A

##### ReportsView States and Transitions

N/A

### Sequence Diagrams

#### AssignShift Diagram:



Figure 7: AssignShift sequence diagram

#### AssignTag Diagram:



Figure 8: AssignTag sequence diagram

#### CreateMember Diagram:



Figure 9: CreateMember sequence diagram

#### CreateSensor Diagram:



Figure 10: CreateSensor sequence diagram

#### CreateTag Diagram:



Figure 11: CreateTag sequence diagram

#### CreateUser Diagram:



Figure 12: CreateUser sequence diagram

#### LoadMap Diagram:



Figure 13: LoadMap sequence diagram

#### Login Diagram:



Figure 14: Login sequence diagram

#### ReceiveMessage Diagram:



Figure 15: ReceiveMessage sequence diagram

#### ViewMemberPosition Diagram:



Figure 16: ViewMemberPosition sequence diagram

#### ViewPathChart Diagram:



Figure 17: ViewPathChart sequence diagram

#### BroadcastText Diagram:



Figure 18: BroadcastText sequence diagram

#### CreateAlert Diagram:



Figure 19: CreateAlert sequence diagram

#### CreateAttendanceReport Diagram:



Figure 20: CreateAttendanceReport sequence diagram

#### CreateReport Diagram:



Figure 21: CreateReport sequence diagram

### Relational Database Schema

#### System Relational Diagram:



Figure 22: Full system relational schema showing how data is saved to the database

## Design Rationale

This section will be used to explain your design from a critical standpoint. This section can also capture good ideas that were abandoned and the reasons for leaving them out of the design.

Use this section to motivate any decisions that will help the reader understand the design that your team is using. No design is perfect. Thus you are being asked to explain the decisions made to support your team’s design. You should concentrate on the tradeoffs you may have made. Here are some guidelines:

A design that is very easy to maintain, may not be the most efficient. Explain this tradeoff in the context of your team’s design.

You may have chosen to store information as opposed to re-generating it on demand (or the other way around.) Why was this decision made? Was it for efficiency? Was it for storage considerations?

When allocating responsibilities among all classes, you may have had a hard time deciding to which specific class assign certain responsibilities. These are important design decisions that need to be explained.

## Review Forms

Review Forms (from each group member) and review Report are appended here.

# Walkthrough Report

# Software Testing Plan

## Introduction

The objective of the tests outlined in this plan is to provide development and QA team members with a framework that can be used to verify all functional and non-functional requirements specified in the SRS document.

## Relationship to other documents

The tests are designed to verify the correct functionality of all class operations described in the SDS and to validate that all functional and non-functional requirements specified in the SRS are present and verified.

## System overview

[This section is focusing on the structural aspects of testing, provides an overview of the system in terms of the components that are tested during the unit test. The granularity of components and their dependencies are defined in this section.]

## Features to be tested/not to be tested

[This section is focusing on the functional aspects of testing, identifies all features and combinations of features to be tested. It also describes all those features that are not to be tested and the reasons for not testing them.]

## Pass/Fail criteria

[This section specifies generic pass/fail criteria for the tests covered in this plan.]

## Approach

[This section describes the general approach to the testing process. It discusses the reasons for the selected integration testing strategy. Different strategies are often needed to test different parts of the system.]

## Suspension and resumption

[This section specifies the criteria for suspending the testing on the test items associated with the plan. It also specifies the test activities that must be repeated when testing is resumed.]

## Testing materials (hardware/software requirements)

[This section identifies the resources that are needed for testing. This should include the physical characteristics of the facilities, including the hardware, software, special test tools, and other resources needed (office space, etc.) to support the tests.]

## Test cases

[This section is the core of the test plan, lists the test cases that are used during testing. Each test case is described in detail in a separate Test Case Specification document. Each execution of these tests will be documented in a Test Incident Report document.]

## Testing schedule

[This section covers responsibilities, staffing and training needs, risks and contingencies, and the test schedule.]

# Gantt Chart

# References

Bandyopadhyay, L. K., S. K. Chaulya, and P. K. Mishra. "Chapters 8 & 11." *Wireless Communication in Underground Mines: RFID-based Sensor Networking*. New York: Springer, 2010. N. pag. Print.

Microsoft. "C# Coding Conventions (C# Programming Guide)." *MSDN*. Microsoft, 2014. Web. 20 Nov. 2014.

Microsoft. "How to Create the Best User Experience for Your Application." *MSDN*. Microsoft, 2014. Web. 20 Nov. 2014.

# Use Case Specification Documents