Software Requirements Specification

For

Attendance Tracking System, Release 1.0

Version 1.0

Prepared by Lee Bell, Graham Kennedy, Jonathan Loudin, Roger Seagle

February 9, 2003

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Everyone	February 9, 2003	Initial Draft	1.0 draft 1
Everyone	February 12, 2003	Revise Draft	2.0 Final Version

1. Introduction

1.1 Purpose
The following document describes the functional and non-functional requirements for the Attendance Tracking System release version 1.0. The contents are intended to be utilized by the software engineering class as guidelines for implementation and testing. This Software Requirements Specification document only covers the main system and does not describe the implementation of the database in which the main system interacts. All the requirements stated in this document are slated for implementation in version 1.0, unless otherwise specified.

1.2 Document Conventions
As of the latest version of this Software Requirements Specification, there are no document conventions.

1.3 Intended Audience and Reading Suggestions
The intended audience is the software engineering class, who will be implementing and testing the Attendance Tracking System. Also, the document is to be utilized by the software engineering professor to evaluate the software's design and features.

1.4 Project Scope
The Attendance Tracking System will allow professors to maintain a record of attendance of students in their respective classes from a Pocket PC. Also, the system will permit the instructor to send a student a notification email of their absence. Furthermore, the program provides different modes to edit attendance, view statistics, and take notes on elements pertaining to attendance. The goal is to provide a professor with an easy, portable solution to attendance record maintenance and attendance statistics.

1.5 References
1. Weigers, Karl. *Software Requirements Specification Template*, http://www.processimpact.com/process_assets/srs_template.doc

2. Overall Description

2.1 Product Perspective
The Attendance Tracking System is intended to replace the manual model of attendance record keeping by means of roll call and paper records. The roll call and paper records are replaced with a single interaction between the professor and the Attendance Tracking System. Professors will be able to view pictures of students on their Pocket PC and quickly maintain attendance records. The features expressed in this Software Requirements Specification document are intended to be fully implemented in version 1.0. The system will be developed in such a way to provide easy addition of enhanced features, which may be desired in subsequent versions.

2.2 Product Features
The main feature of the Attendance Tracking System is for a professor to take roll from the Pocket PC. Pictures will be displayed for each student in the class, so the professor can recognize students without having to manually call roll. If students are absent, the system sends a notification email to the student stating the current number of absences the student has accumulated. Also, the system allows the professor to view pertinent statistics on student's attendance record for the specified class. Finally, another feature of the Attendance Tracking System is the ability to easily edit the attendance record for any given day.

2.3 User Classes and Characteristics
Professors — A professor is a faculty member of Wake Forest University in Winston Salem, North Carolina. There are 357 faculty members at the university and it is estimated that almost all faculty members will utilize the Attendance Tracking System. Professors will have

multiple classes and interaction with the system will occur at multiple times throughout the day from multiple professors. The professor should be allowed to take, edit, and view attendance records at any desirable time. All professors have wireless Internet access through their Pocket PC's while on campus. This connection provides the backbone of the Attendance Tracking System.

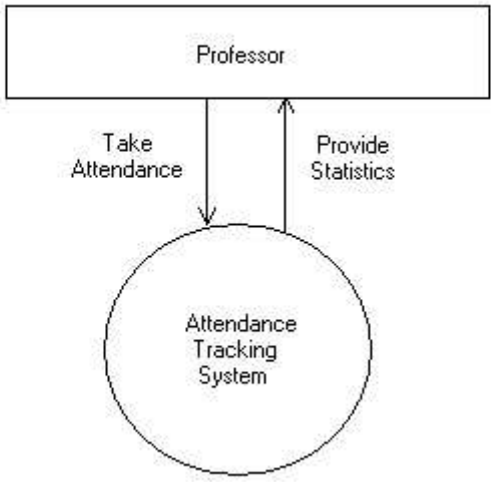

Figure 1: Context diagram for Attendance Tracking System.

2.4 Operating Environment

OE-1: The Attendance Tracking System shall function on the Pocket PC provided by the university. This entails the system to operate on the Windows CE platform.

OE-2: The Attendance Tracking System shall interface between Windows CE and a Linux server designated to store the attendance records. The Web Server and Database Software have not been established at this point. Updated versions of this document will include information pertaining to these specific operating environments.

OE-3: The Attendance Tracking System will operate with a yet to be established email client for notification of absences sent to students.

2.5 Design and Implementation Constraints

CO-1: The time allotted for this project will be limited to the Spring 2003 semester.

CO-2: The language for the project will be Visual C++ and the development environment will be the embedded Visual C++ 3.0 environment.

CO-3: All the HTML code for the user manual will conform to the HTML 4.0 standard.

2.6 User Documentation

UD-1: The system will provide an online user manual in HTML that describes the functionality and options available to the user.

UD-2: The system will provide a hard copy of the user manual, which is identical to the manual outlined in UD-1

2.7 Assumptions and Dependencies

AS-1: The database mentioned within this Software Requirements Specification document is previously administered with the correct information needed by the Attendance Tracking System.

AS-2: For attendance policies, since professors do not usually tabulate tardiness, it is assumed a student is either present or absent. The notes functionality allows the professor the option to comment on tardiness and the edit mode allows the professor to correct attendance due

to tardiness.

DE-1: Class data being used for setup and student recognition is dependent on information in a database administered outside of the capabilities of the Attendance Tracking System.

DE-2: Statistics on student attendance is dependent on a professor's consistent utilization of the system for each class period.

3. System Features

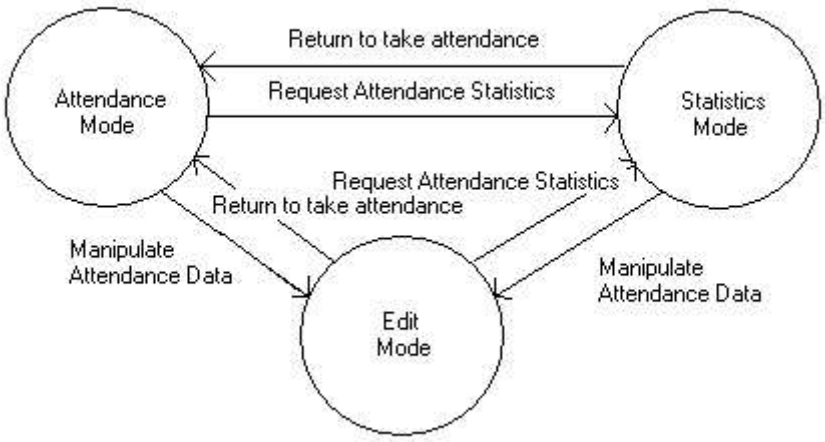

Figure 2: State Diagram for transition between modes within the system.

3.1 Roll Mode

3.1.1 Description and Priority

A user whose identity has been verified will be able to retrieve a roster and take roll at class time. The Pocket PC will determine the correct roster for the current date and time, retrieve that roster from a database server, and display it for the purposes of taking roll. The user will be able to designate each class member as present or absent, and upload that status information to the server. The server will notify the class members marked as absent by e-mail. Priority = High.

3.1.2 Stimulus/Response Sequences

Stimulus: User loads system.
Response: System queries server and if a class exists for the current date, time, and user, then that roster is retrieved and displayed.

Stimulus: User requests that a class member be marked absent.
Response: The class member is designated as absent.

Stimulus: User requests that a class member be marked present.
Response: The class member is designated as present.

Stimulus: User requests that the data be sent to the server for storage.
Response: The absence data for the current roster are transmitted to the server.

Stimulus:	Server receives data indicating a class member has been recorded absent.
Response:	E-mail is sent to that class member.

3.1.3 Functional Requirements

Roll.Retrieve.Time:	The system shall retrieve the current time.
Roll.Retrieve.Date:	The system shall retrieve the current date.
Roll.Retrieve.Roster:	The system shall retrieve the roster from the server that matches the current time and date.
Roll.Retrieve.Roster.No:	If no roster is found to match current time and date, default menus will be displayed.
Roll.Display.Roster:	The system will display the roster on screen.
Roll.Display.Members:	The roster will display the pictures and names of class members.
Roll.Mark.Present:	Members have the ability to be marked as present.
Roll.Mark.Absent:	Members have the ability to be marked as absent.
Roll.Transmit.Data:	The system will transmit the status of each class member in the active roster to the server.
Roll.Notify.E-mail:	The system will notify all class members by e-mail if they are marked as absent along with their current status within the attendance policy.

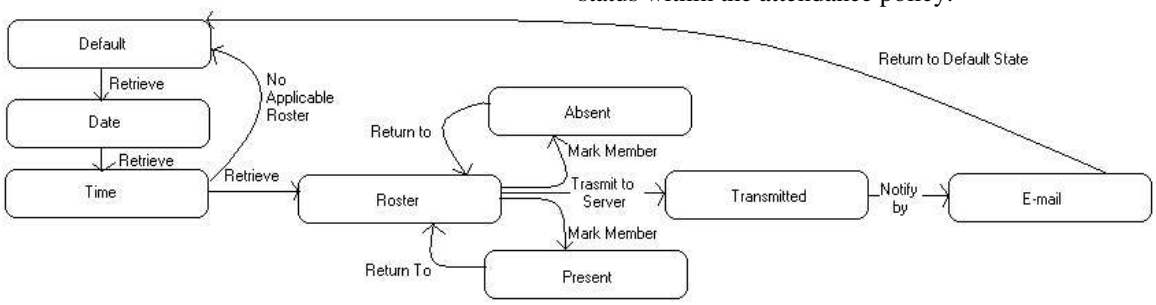

Figure 3: State Transitions within *Roll Mode*.

3.2 Edit Mode

3.2.1 Description and Priority

A user whose identity has been verified who wishes to change the absence status of one or more class members as is related to a past class period will have the ability to retrieve the roster data for that class period from the server. Absence data may then be adjusted and the corrections uploaded to the server. Any new absences will result in notification by e-mail to the class member in question. A change from absent to present will result in a notification that an absence has been removed. Priority = High.

3.2.3 Stimulus/Response Sequences

Stimulus:	The user requests to view absence data from a certain date and time.
Response:	The roster and absence data are retrieved from the server and displayed.
Stimulus:	The user requests that a class member's status be changed to absent.

Response:	The class member's status is changed to absent.
Stimulus:	The user requests that a class member's status be changed to present.
Response:	The class member's status is changed to present.
Stimulus:	The user indicates that they are finished making changes.
Response:	The altered data is uploaded to the server and server data is updated to reflect the changes. E-mail notification is sent to the class members affected.

3.2.3 Functional Requirements

Edit.Retrieve.Time:	The system shall allow the user to choose the date of the roster to be retrieved.
Edit.Retrieve.Date:	The system shall allow the user to choose the date of the roster to be retrieved.
Edit.Retrieve.Roster:	The system will retrieve the roster and absence data from the requested date and time.
Edit.Retrieve.Roster.No:	If no such roster exists, an error message will be displayed.
Edit.Display.Roster:	The system will display the roster on screen.
Edit.Display.Members:	The roster will display the pictures and names of class members along with current absence status on screen.
Edit.Mark.Present:	Members will have the ability to have their status changed to present.
Edit.Mark.Absent:	Members will have the ability to have their status changed to absent.
Edit.Transmit.Data:	The system will transmit the new status of each class member in the active roster to the server.
Edit.Notify.Absent:	The system will notify all class members by e-mail if their status has been changed to absent along with their current status within the attendance policy.
Edit.Notify.Present:	The system will notify all class members by e-mail if their status has been changed to present along with their current status within the attendance policy.

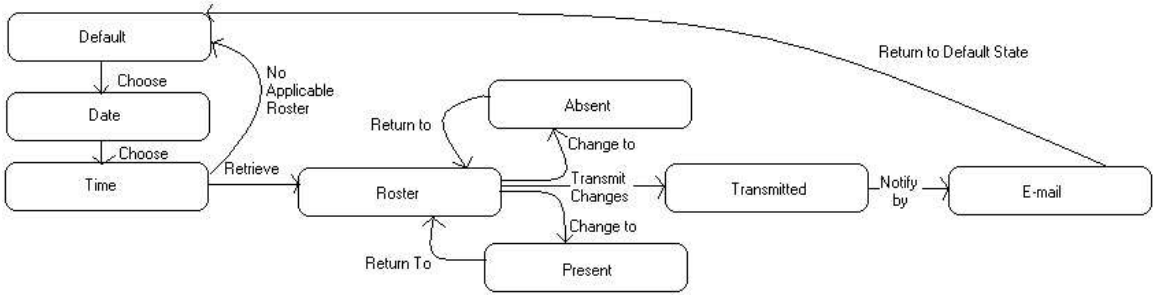

Figure 4: State Transitions within *Edit Mode*.

3.3 Statistics Mode

3.3.1 Description and Priority

A user whose identity has been verified may request absence statistics for class members in classes under their control. Statistics for an individual class member may be retrieved and displayed with regard to number, frequency, date of absences as well as the percentage of classes missed over a user-defined period of time. A list of class members who were absent for a chosen class period may also be retrieved. Priority = High.

3.2.3 Stimulus/Response Sequences

Stimulus:	User requests a statistical record.
Response:	Statistics are calculated, retrieved from the server, and displayed.

3.3.3 Functional Requirements

Stats.Define.Range:	The user shall have the ability to set the date or range of dates for which statistics will be retrieved.
Stats.Request.Number:	The user shall have the ability to request the number of absences for a class member.
Stats.Request.Dates:	The user shall have the ability to request the dates of the absences for a class member.
Stats.Request.Frequency:	The user shall have the ability to request the frequency of absences for a class member.
Stats.Request.Percentage:	The user shall have the ability to request the percentage of class periods that a class member was absent.
Stats.Request.List:	The user shall have the ability to request a list of the class members absent on a given date.
Stats.Retrieve.Number:	The system shall have the ability to calculate and retrieve the number of absences for a class member from the server.
Stats.Retrieve.Dates:	The system shall have the ability to calculate and retrieve the dates of the absences for a class member from the server.
Stats.Retrieve.Frequency:	The system shall have the ability to calculate and retrieve the frequency of absences for a class member from the server.
Stats.Retrieve.Percentage:	The system shall have the ability to calculate and retrieve the percentage of class periods that a class member was absent from the server.
Stats.Retrieve.List:	The system shall have the ability to retrieve a list of the class members absent on a given date from the server.
Stats.Display.Number:	The system shall have the ability to display the number of absences for a class member.
Stats.Display.Dates:	The system shall have the ability to display the dates of the absences for a class member.

Stats.Display.Frequency:	The system shall have the ability to display the frequency of absences for a class member.
Stats.Display.Percentage:	The system shall have the ability to display the percentage of class periods that a class member was absent.
Stats.Display.List:	The system shall have the ability to display a list of the class members absent on a given date.

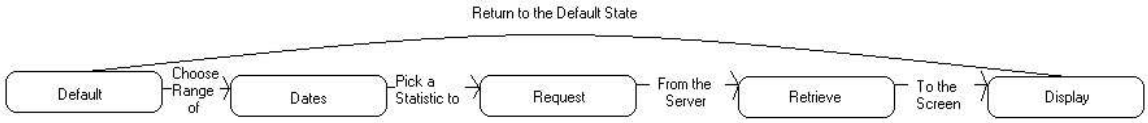

Figure 5: State Transitions within *Statistics Mode*.

4. External Interface Requirements

4.1 User Interfaces

UI-1:	The Attendance Tracking System shall provide pictures above the names of students in the class to aid in taking roll.
UI-2:	These pictures can be clicked with a stylus in order to indicate whether a student is present or absent.
UI-3:	All modifications to the database will be done through a stylus, the keyboard would only be needed to enter passwords.
UI-4:	The Attendance Tracking System will provide a help link that will download a user manual and project it onto the screen in case the user has difficulty in using the program.
UI-5:	The program will provide a page that produces current statistics on class attendance.

4.2 Hardware Interfaces

HI-1:	The Attendance Tracking System will reside on the Pocket PC, however, it will be able to report its data to a database set up on a Solaris machine.
HI-2:	The Attendance Tracking System will also be able to tell the Solaris machine to send e-mails to students who have missed class.
HI-3:	The Pocket PC will use a wireless Ethernet card to communicate with the database set up on the Solaris machine.

4.3 Software Interfaces

SI-1:	Attendance Tracking System
SI-1.1:	This software will transmit the attendance of a class to a database on a Solaris machine via wireless Ethernet.
SI-1.2:	The user will be allowed to modify attendance records at any time.
SI-1.3:	If the user forgets to transmit the information, the system will automatically send it for them at the end of the class.
SI-2:	Database The Attendance Tracking System will communicate with the database to perform the following options.
SI-2.1:	To allow a user to enter attendance.
SI-2.2:	To allow a user to modify attendance.

SI-2.3: To allow a user to query a system to gain statistics concerning individual and class attendance.

SI-3: Mail Message Generator
If a student is absent, the database will send the necessary information to this interface and an e-mail message will be sent to the student.

4.4 Communications Interfaces

CI-1: The security of a user must be consistent through the use of passwords.
CI-2: The Attendance Tracking System will signal the mail message generator when it needs to send an email to the student.
CI-3: The Attendance Tracking System will communicate to the database through a wireless Ethernet system.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

PE-1: The program must be able to be run concurrently by multiple professors. During peak times of usage (9:00 AM to 2:00 PM), the system shall support approximately 20 users.
PE-2: Transmission of roll data shall occur in under 5 seconds using the wireless network card.
PE-3: Acknowledgement of roll received (confirmation) shall be returned within 8 seconds.
PE-4: Queries upon the database shall be performed in less than 5 seconds.
PE-5: Upon start of the roll program, roll information shall be displayed on the instructor's Pocket PC within 10 seconds.
PE-6: Email messages to absent students shall be sent within one hour of the conclusion of a class.
PE-7: The program shall support taking roll for class sizes of up to 100 students. With a maximum class size, performance must still conform to all performance requirements.

5.2 Safety Requirements

No safety requirements have been identified.

5.3 Security Requirements

SE-1: An instructor shall permitted to view and edit absence information in the database for only his/her classes.
SE-2: In order to take roll or edit database information, the instructor shall be required to enter a password. This password shall be stored on the instructor's Pocket PC after initial entry in order that it must only be entered once.
SE-3: Passwords shall be stored in an instructor database and verified upon each session of roll or database modification.
SE-4: An instructor shall be allowed to change his/her password only by supplying his/her existing password. The updated password shall be stored in the instructor database and on the instructor's Pocket PC.

5.4 Software Quality Attributes

Availability-1: The system shall be available to all instructors through Information Systems during their normal hours of operation.

Reliability-1: Due to the use of a wireless network, reliability of the system at all times is not guaranteed. However, overall reliability of the system and roll information shall be achieved through the process of database manipulation.

Reusability-1: The system shall be able to be reused for each new semester.

Robustness-1: If no network connection can be established to receive the roll information, the instructor shall be allowed to enter roll later.

Robustness-2: If a network connection is lost during roll, the program shall allow the instructor to transmit roll information at a later time.

Updatability-1: The system shall allow for addition or deletion of instructors, students, and classes while incorporating new semesters.

Usability-1: Usability of the system shall be achieved through an online help pages and an introductory training session for all instructors upon installation of the program on their Pocket PCs.

6. Other Requirements

6.1 Database Requirements

DA-1: The system shall include three databases: *student*, *class*, and *instructor*.

DA-2: The *student* database shall contain student information including name, email address, and student identification number.

DA-3: The *class* database shall contain information about classes including class title, department code and course number, meeting times, students enrolled, and absences. The *class* database shall be grouped by department code and course number.

DA-4: The *instructor* database shall contain information about instructors including name, email address, password, and classes taught.