



US 20060294552A1

(19) **United States**(12) **Patent Application Publication**
Swanson et al.(10) **Pub. No.: US 2006/0294552 A1**(43) **Pub. Date: Dec. 28, 2006**(54) **AUDIENCE RESPONSE SYSTEM AND METHOD****Publication Classification**(75) Inventors: **Mark R. Swanson**, Wisconsin Rapids, WI (US); **William L. Moss JR.**, Wisconsin Rapids, WI (US); **Peter William Jungwirth**, Wisconsin Rapids, WI (US)(51) **Int. Cl.****H04N 5/44** (2006.01)**G06F 13/00** (2006.01)**H04N 7/18** (2006.01)**G06F 3/00** (2006.01)**H04N 5/445** (2006.01)(52) **U.S. Cl.** **725/80**; 725/81; 348/734; 725/39

Correspondence Address:

GARDNER CARTON & DOUGLAS LLP
ATTN: PATENT DOCKET DEPT.
191 N. WACKER DRIVE, SUITE 3700
CHICAGO, IL 60606 (US)(73) Assignee: **Renaissance Learning, Inc.**, Wisconsin Rapids, WI(21) Appl. No.: **11/474,838**(22) Filed: **Jun. 26, 2006****Related U.S. Application Data**

(60) Provisional application No. 60/694,414, filed on Jun. 27, 2005. Provisional application No. 60/729,428, filed on Oct. 21, 2005.

(57) **ABSTRACT**

An audience response system is provided. In various implementations, the response system includes a base station, handheld units, and a wireless network. The response system may have a prompted mode and an unprompted mode of operation. In the prompted mode of operation, the base station transmits questions to the handheld units over the wireless network. The questions are displayed to users on their handheld units. The users respond to the question, and their responses are transmitted to the base station via the wireless network. In the unprompted mode, answer templates are transmitted and displayed to the users, with the actual questions being conveyed in some other manner, such as verbally or by written handout.

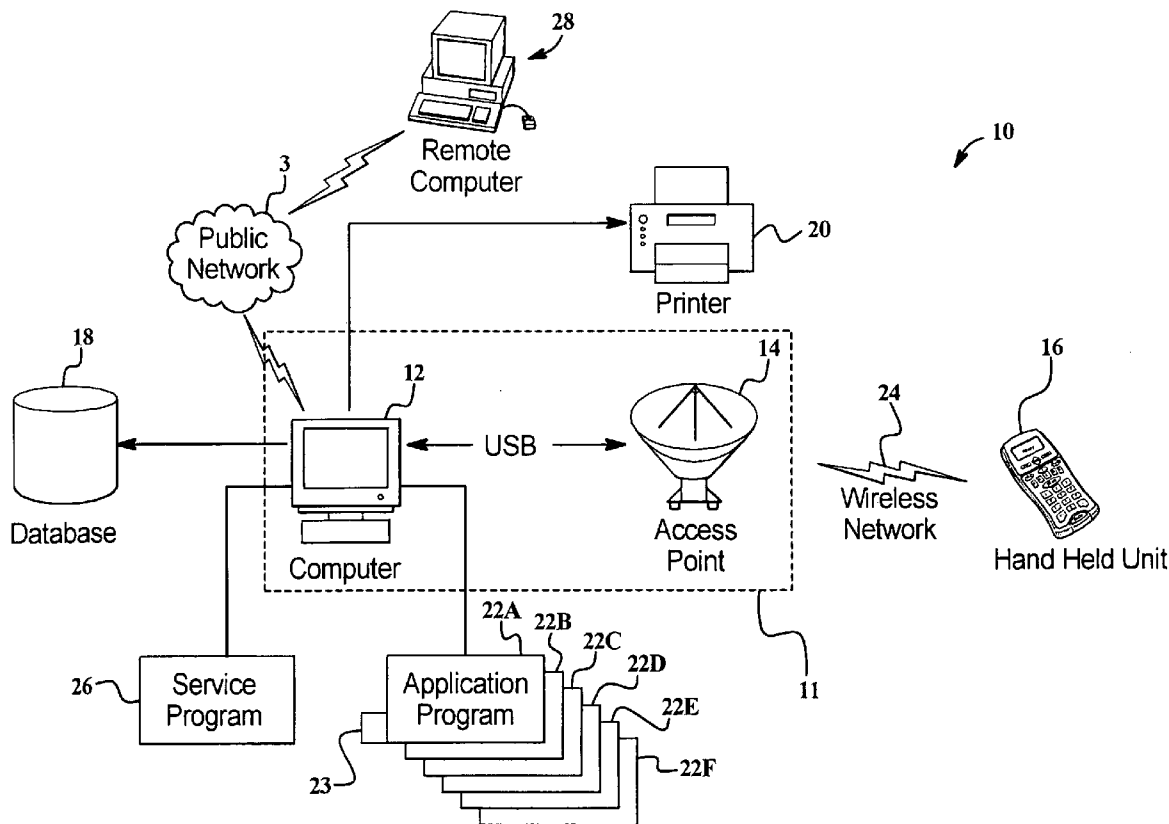


FIG. 2

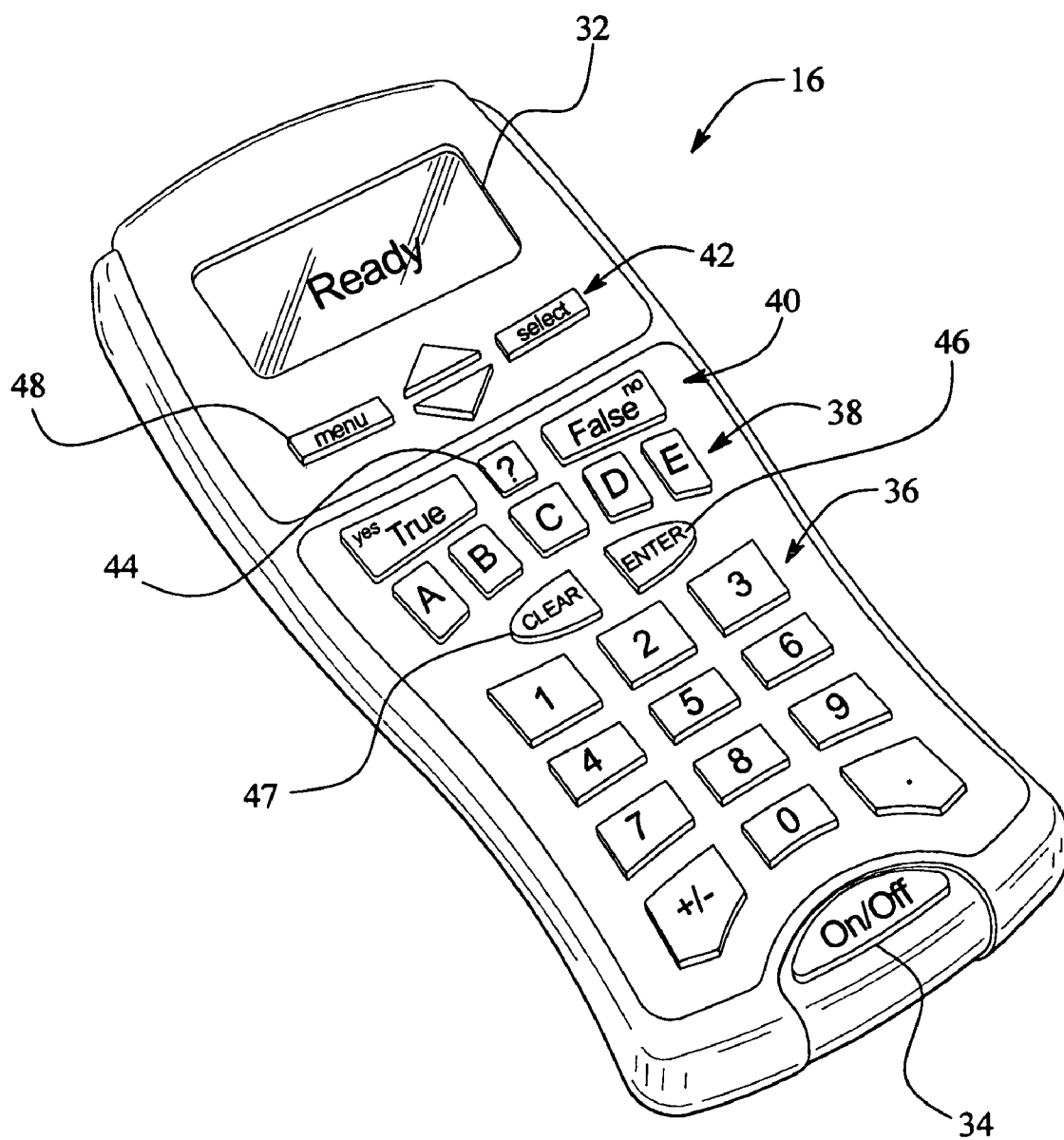


FIG. 3

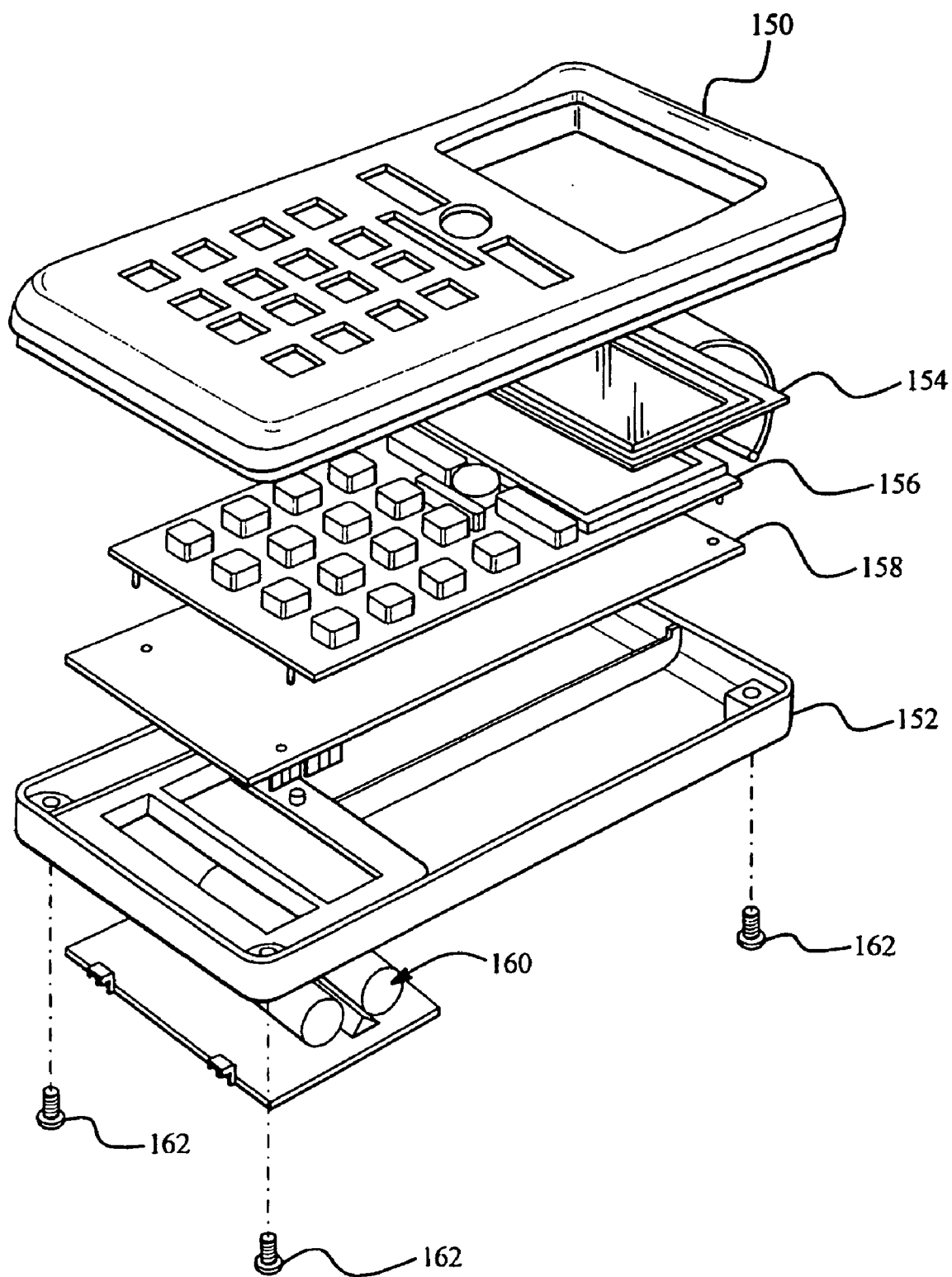


FIG. 4

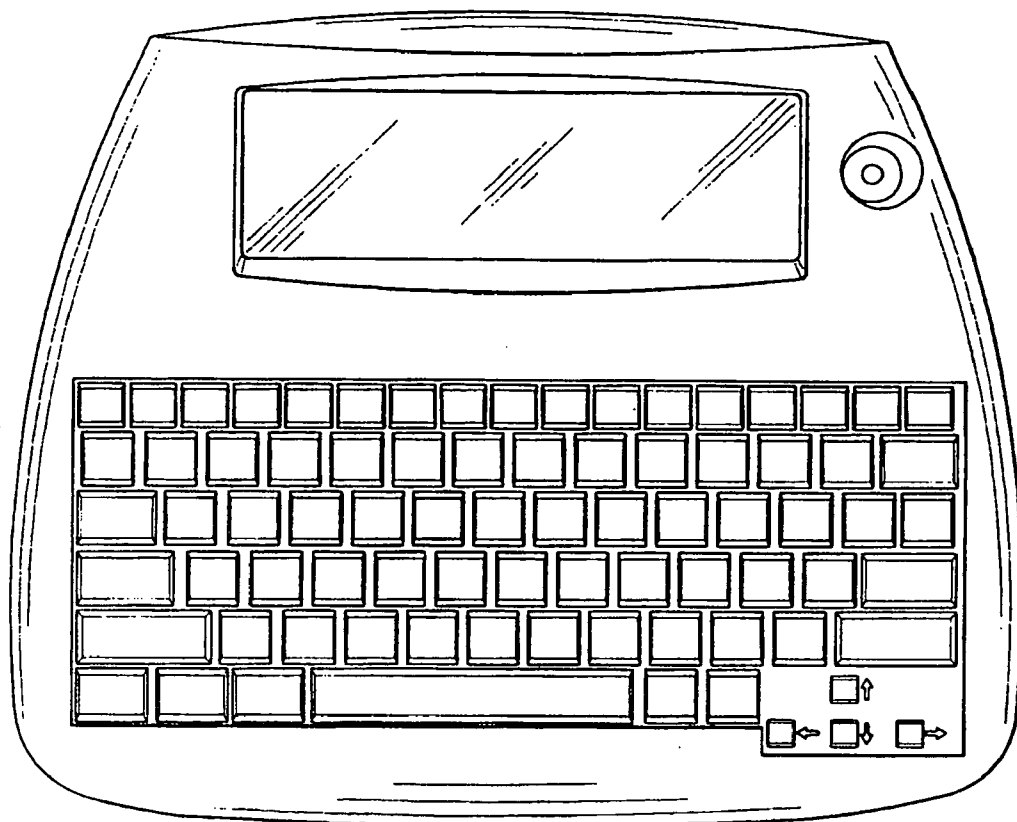
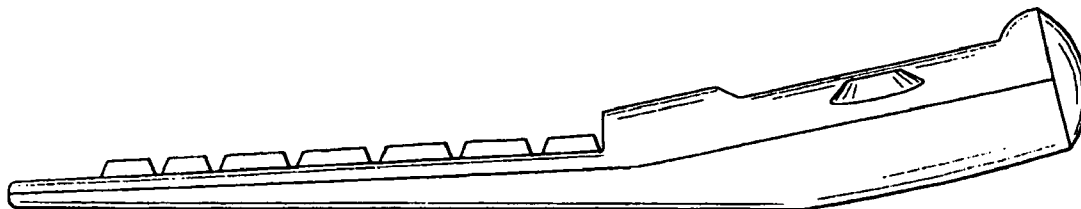


FIG. 5



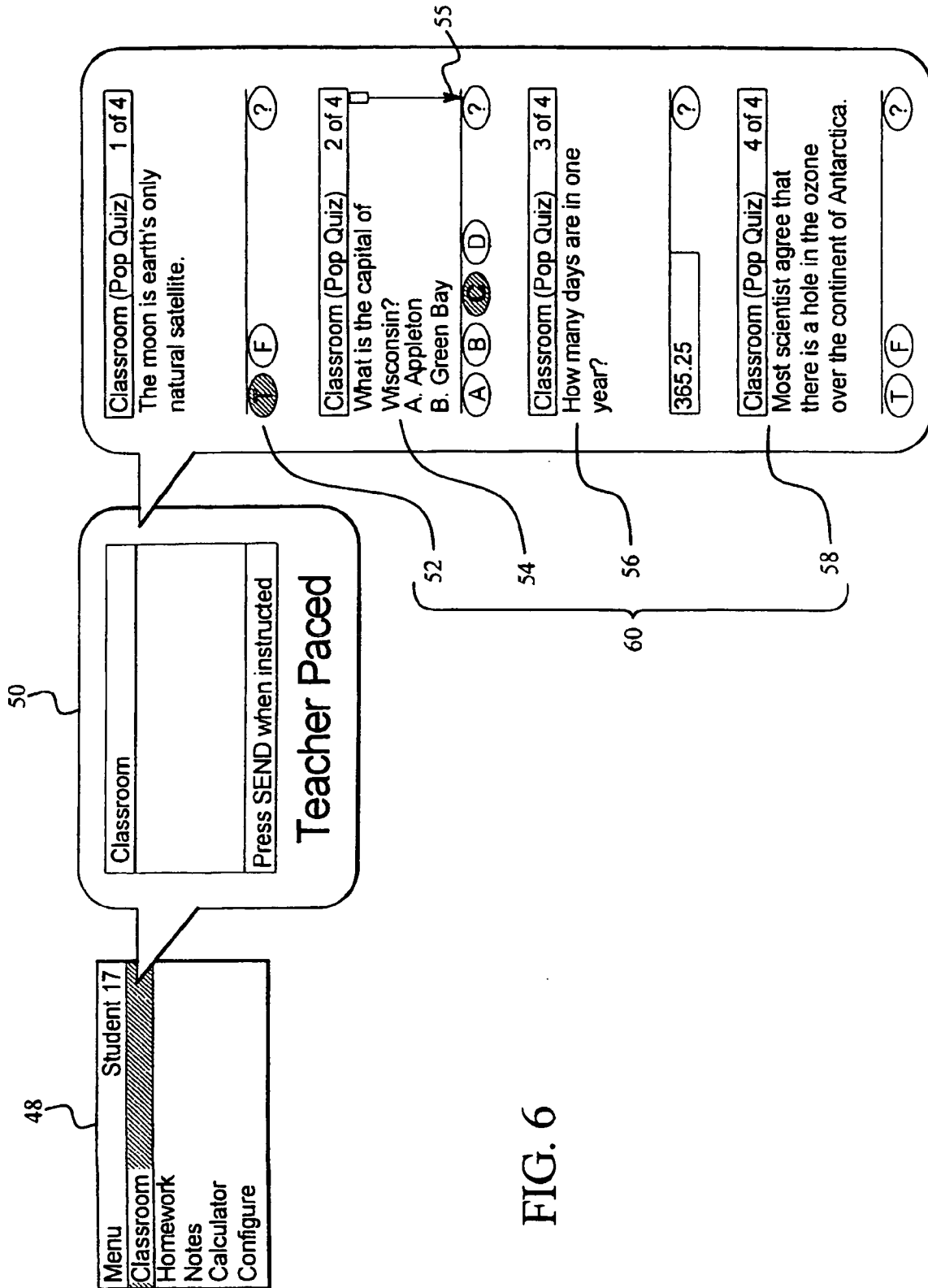


FIG. 6

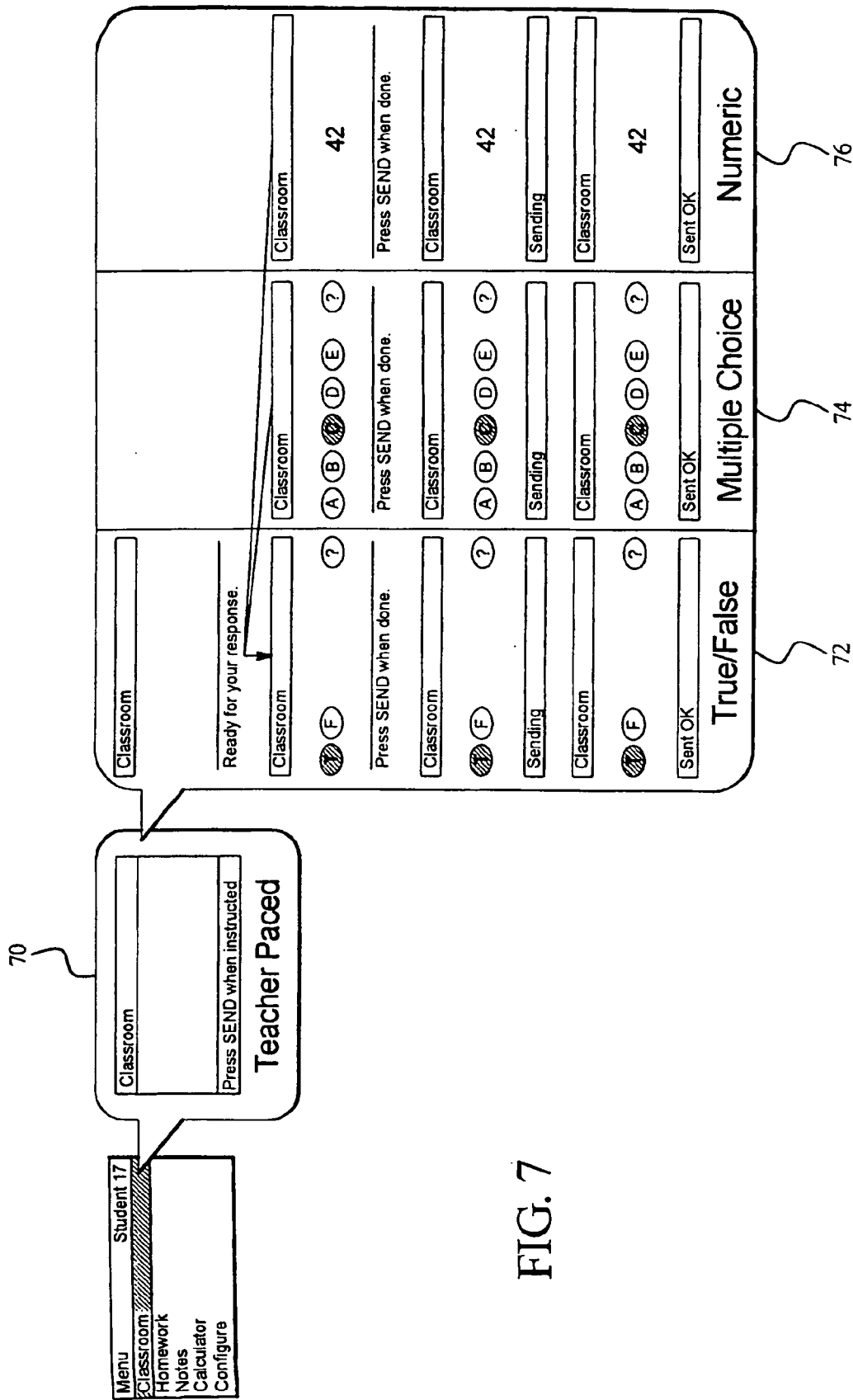


FIG. 7

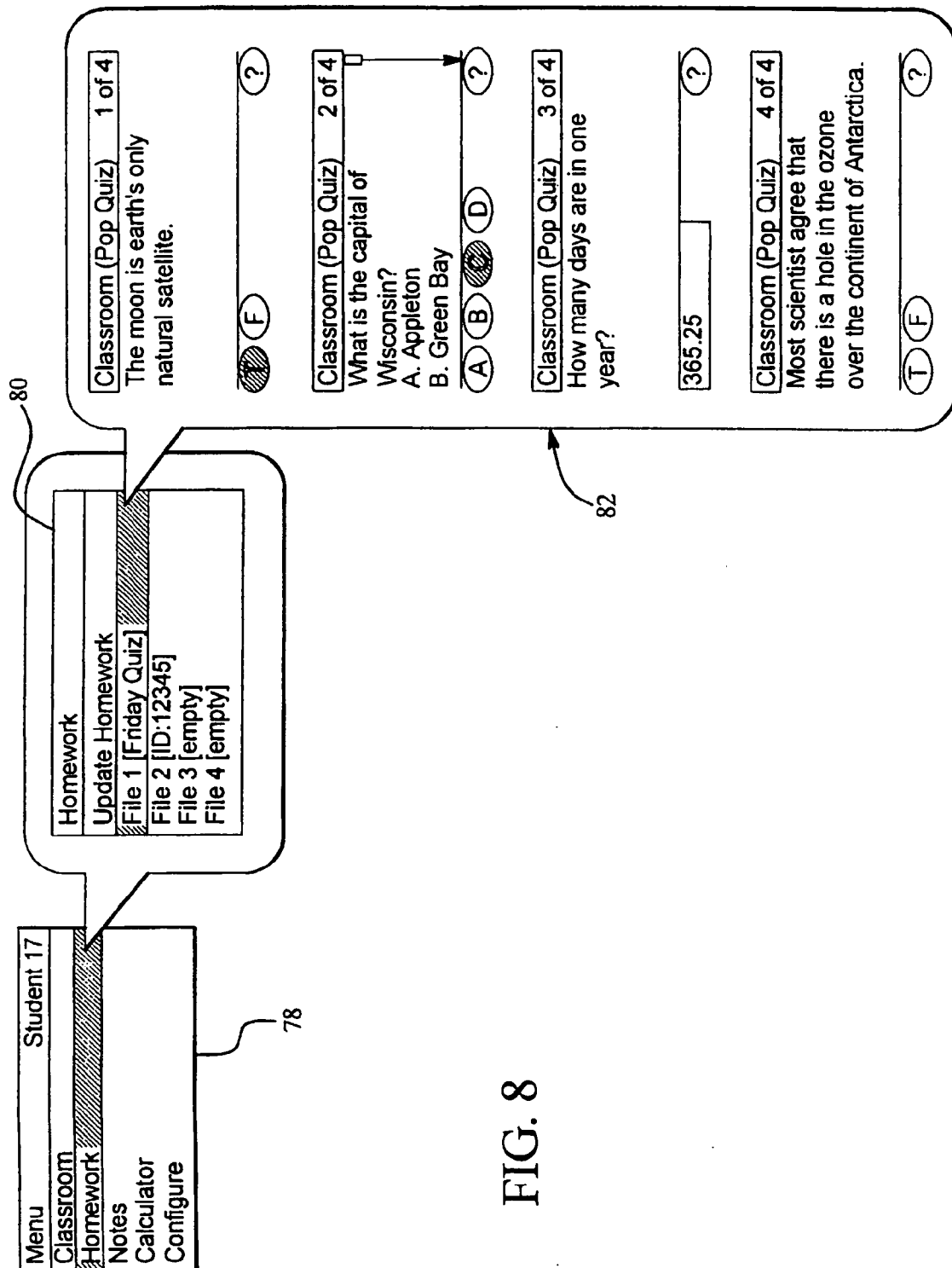


FIG. 8

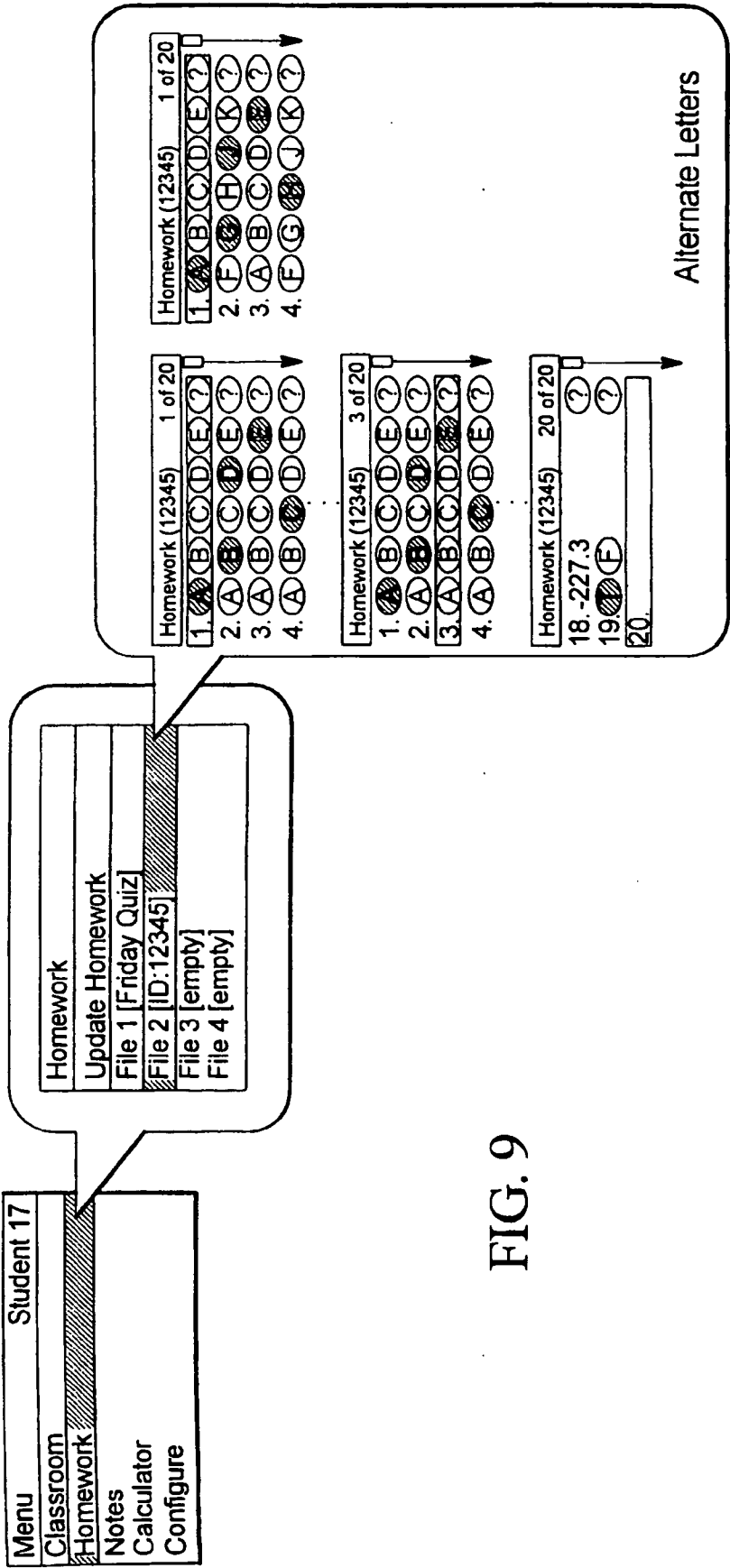


FIG. 9

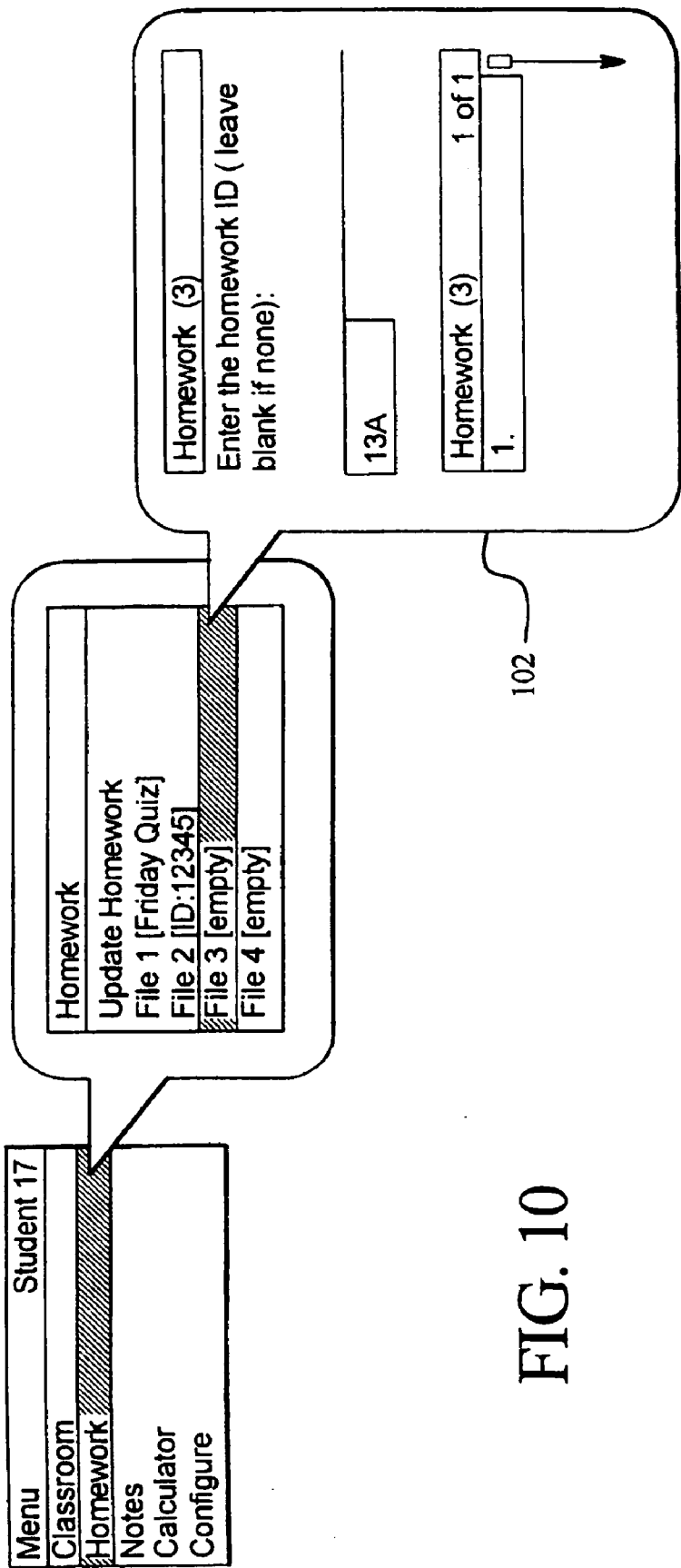


FIG. 10

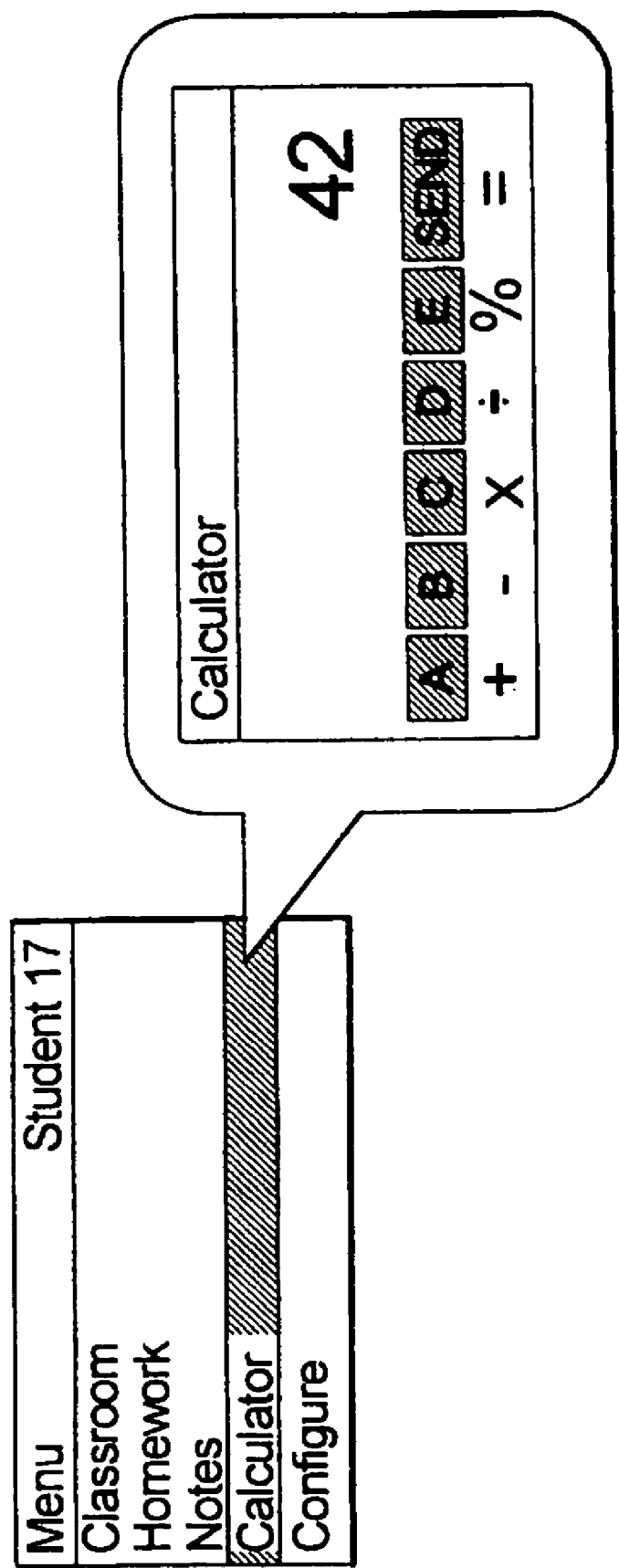


FIG. 11

AUDIENCE RESPONSE SYSTEM AND METHOD

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of the filing date of U.S. Provisional Application No. 60/694,414, filed Jun. 27, 2005 and U.S. Provisional Application No. 60/729,428, filed Oct. 21, 2005, each of which are incorporated herein by reference in their entirety.

FIELD OF THE INVENTION

[0002] The invention relates generally to wireless audience response systems, and, more particularly, to wireless audience response systems that solicit responses to questions posed to an audience by wireless communication with handheld units possessed by members of the audience.

BACKGROUND OF THE INVENTION

[0003] When a presentation is given to an audience, whether it is a lecture to a class of students or a sales pitch to a group of executives, it is often difficult to gauge whether the audience is absorbing or even paying attention to the information being presented. Audience response systems address this difficulty. Many audience response systems include individual handheld units, each with several input keys. In such systems, each audience member has one of the handheld units, and, in response to a question spoken by the presenter, each audience member presses a button corresponding to the appropriate response. In most existing systems, however, the handheld units do not have display screens. Furthermore, even those that have display screens lack the ability to switch between a prompted mode, in which questions are electronically transmitted to handheld units and displayed on the screen, and an unprompted mode, in which only an answer template is transmitted to and displayed on the handheld units, with the question itself being conveyed to the audience verbally or in written form.

BRIEF SUMMARY OF THE INVENTION

[0004] In accordance with the foregoing, an audience response system is provided. In various embodiments of the invention, the response system includes a base station, handheld units, and a wireless network. The response system may have a prompted mode and an unprompted mode of operation. In the prompted mode of operation, the base station transmits questions to the handheld units over the wireless network. The questions are displayed to users on their handheld units. The users respond to the question, and their responses are transmitted to the base station via the wireless network. In the unprompted mode, answer templates are transmitted and displayed to the users, with the actual questions being conveyed in some other manner, such as verbally or by written handout.

[0005] There are a variety of possible implementations of the response system. In some implementations, the question or questions are transmitted to users as part of one or more markup languages documents. The handheld units process the markup language documents and display the questions and appropriate answer fields to the users in accordance with the markup language documents. When used in an educational setting, the markup language documents may also contain homework assignments as well as notes to the parents or guardians of the users.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] FIG. 1 illustrates a wireless audience response system according to an embodiment of the invention;

[0007] FIGS. 2 & 3 illustrate a handheld unit that may be used in an embodiment of the invention;

[0008] FIGS. 4 & 5 illustrate another handheld unit that may be used in an embodiment of the invention;

[0009] FIG. 6 illustrates the sequence of displays shown on a handheld unit for a pop quiz that is given by an audience response system, in which the response system is operating in a teacher-paced, prompted mode;

[0010] FIG. 7 illustrates the sequence of displays shown on a handheld unit for a quiz that is given by an audience response system, in which the response system is operating in a teacher-paced, unprompted mode;

[0011] FIG. 8 illustrates the sequence of displays shown on a handheld unit for a homework assignment, in which the response system is operating in a student-paced, prompted mode;

[0012] FIG. 9 illustrates the sequence of displays shown on a handheld unit for a homework assignment, in which the response system is operating in a student-paced, unprompted mode with multiple choice answer templates;

[0013] FIG. 10 illustrates the sequence of displays shown on a handheld unit for a homework assignment, in which the response system is operating in a student-paced, unprompted mode with fill-in-the blank answer templates; and

[0014] FIG. 11 illustrates a calculator function of a handheld unit according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0015] The architecture and operation of a wireless audience response system configured according to an embodiment of the invention will now be described. Referring to FIG. 1, the response system, generally labeled 10, includes a base station 11 and a plurality of handheld units 16. The base station 11 includes a computer 12 and a wireless access point (AP) 14. The computer 12 is communicatively linked to the AP 14 via universal serial bus (USB), while the AP 14 is communicatively linked to the handheld units 16 via a wireless network 24. In one embodiment, the wireless network 24 is a personal area network (PAN) operating according to the IEEE 802.15.4 standard, with the AP 14 acting as a PAN coordinator. The system 10 also includes a database 18 on which data representing questions and their corresponding answers are stored. The system 10 also includes a printer 20 communicatively linked to the computer 12, which allows an instructor or other speaker to print out results of tests or surveys taken by users of the handheld units 16, first through sixth application programs 22A-22F (generically referred to as application programs 22) executing on the computer 12, which dictate how the computer 12 administers the questions transmitted to the handheld units 16 and processes the answers received in response thereto, and a service program 26, which provides communication and routing services to the application programs 22. Finally, the system 10 may be communicatively linked with a remote

computer 28 via a public network (e.g., the Internet). The remote computer 28 hosts a web site from which questions may be downloaded.

[0016] Referring to FIG. 2, an example of how the handheld unit 16 may be implemented will now be described. It is understood, however, that other configurations are possible, such as the full-keyboard configuration shown in FIGS. 4 & 5. In the markup language examples contained herein, the moniker “Wasabi” will sometimes be used to refer to the handheld unit implementation of FIG. 2. The handheld unit 16 is designed to be inexpensive, lightweight, rugged, and simple to use. It has a graphical liquid crystal display 32 and twenty-five keys to support multiple response types. The keys include an ON/OFF key 34, a group 36 of numeric keys, a group 38 of multiple choice keys, a group 40 of positive/negative keys, a group 42 of navigation keys, and group of immediate keys, including an ENTER key 46, a CLEAR key 47, an ON/OFF key 34, and a menu key 48. The ON/OFF key 34 is configured so as to reduce the chance of accidental pressing (having, for example, raised ribs on the enclosure or being aligned just below flush with the enclosure). As shown in FIG. 2, the keys are grouped into functional areas. For example, the group 38 of multiple choice keys A thru E is commonly used when responding to multiple choice questions. The group 36 of numeric keys, 0 thru 9, is used when responding to a question requiring a numeric reply. Each group of keys has elements that indicate to the user that the keys are, in fact, part of the same group. Possible ways to accomplish this include, but are not limited to key style (shape, color, label text font/size/color, etc.) and separators in enclosure (altitude, channels, borders, etc.)

[0017] Referring to FIG. 3, the handheld unit 16, in an embodiment of the invention, includes an enclosure having a top portion 150 and a bottom portion 152. The bottom portion 152 has a recess on its outer surface, in which a pair 166 of batteries is housed. In various embodiments, the enclosure is made of rugged, lightweight, low cost plastic (e.g., polycarbonate/abs blend). The finish is textured, non-slip, and fingerprint resistant. The enclosure also prevents battery removal without the use of a tool to reduce the choking hazard. In one embodiment, the enclosure has an envelope of about 5.25 inches by about 1.8 inches by about 0.8 inches.

[0018] Disposed within the enclosure are an LCD unit 154, an elastomeric keypad 156, and a printed circuit board (PCB) 158. Disposed on the PCB 158 is a low-end microcontroller with adequate resources to support the LCD unit 154, the keypad 156 and the wireless communication functions of the handheld unit 16. Also disposed on the PCB 158 is a wireless communication circuit, such as a Chipcon CC2420 802.15.4 radio (Reduced Function Device). The power source for the components on the PCB 158, the keypad 156 and the LCD unit 154 is supplied by the pair 166 of batteries, which may be two standard AA (R6) alkaline batteries. Each handheld unit 16 has a unique identification number, which is electronically stored therein. In addition to providing electrical continuity between circuits, the PCB 158 provides at least two specialized functions. First, it incorporates the antenna for the wireless communication circuit. Second, it incorporates the appropriate landing zones for the carbon conductors on the back of the keypad 156.

[0019] The operation of the handheld unit 16 according to an embodiment of the invention will now be described. The general usage pattern is as follows: the user receives a question, which is either displayed on the LCD screen 32, or provided some other way, such as verbally or written. The user presses one or more keys to formulate a response. In one embodiment, key action is invoked on key closure. The user then presses the ENTER key 46 to transmit the response wirelessly to the base station 11. The handheld unit 16 has a response queue acts as a holding area for responses. The content of this queue changes when keys are pressed and is displayed prominently on the LCD 32. However, the actual data stored in the queue and transmitted is not necessarily what the user sees on the LCD screen 32. For example, when LCD 32 displays “True,” a single data byte can be used to represent a positive response. This is done to minimize memory use and data packet size. The maximum response size is based on the size of the LCD screen 32, the font size, and/or the amount of memory available in the microcontroller of the handheld unit 16.

[0020] The functions of individual keys and groups of keys shown in FIG. 2 according to an embodiment of the invention will now be described. Pressing the ON/OFF key 34 with no other keys down toggles the device state (on or off). Pressing the CLEAR key 47 causes a response queue to be cleared. The response queue is also cleared when the handheld unit 16 is powered on.

[0021] When the ENTER key 46 is pressed, the contents of the response queue are transmitted via the wireless communication circuit. During transmission, the text “sending” is displayed on the LCD screen 32 until the next question is received from the base station 11.

[0022] The multiple choice keys group 38 is used to enter single key responses. When one of the keys of this group is pressed, the handheld unit 16 replaces the contents of the response queue. The group 40 of positive/negative keys is similarly used to enter single key responses. As with the multiple choice keys, pressing one of the positive/negative keys causes the handheld unit 16 to replace the contents of the response queue. In addition to indicating an answer of “true” or “false,” the positive/negative keys are used for Yes/No responses. The handheld unit 16 also includes a “?” key 44, which may be used as a “help” key.

[0023] The group 36 of numeric keys is used to enter multi key responses. Positive, negative, whole, and decimal numbers may be entered. If the response queue contains a multiple choice or a positive/negative response when a numeric key is pressed, an implied CLEAR is executed first. When a key of the numeric key group 36 is pressed, the associated value is added to the rightmost position of the response queue, unless the queue is full, in which case it is ignored.

[0024] In an embodiment of the invention, the handheld unit 16 manages power as follows. The microprocessor sleeps as much as possible (between keypresses) to maximize device runtime. Furthermore, the handheld unit 16 automatically powers off after 30 minutes of inactivity, regardless of key states. A continuously held down key does not constitute activity. If one or more keys are held down accidentally by something sitting on the device or during storage or transport, the handheld unit 16 will power itself

off when the timeout interval expires, and will not power on again until all keys are released and only the ON/OFF key 34 is pressed.

[0025] In one embodiment of the invention, the handheld unit 16 uses a virtual “key chain” having one or more “keys.” The virtual key chain (or simply “key chain”) stores authentication information for a particular user. Each “key” may be used to authenticate the user (and the handheld unit 16) to one of the application programs 22. The virtual key chain is stored in a memory of the handheld unit 16. The purpose of the virtual key chain is to avoid forcing the user to reauthenticate every time the user wishes to interact with one of the application programs 22.

[0026] Referring back to FIG. 1, the operation of the application program 22 according to an embodiment of the invention will now be described. The application program 22 initiates one or more “activities.” As used herein, an “activity” is made up of one or more discrete tasks referred to as “actions.” An activity can be a single action or a complex sequence of actions. For example, an activity can be something as simple as displaying a message to one of the handheld units 16, or an activity can be something more sophisticated such as requesting a response to a multiple-choice question. An activity can have a static design, such as a quiz that includes 10 actions, each action being a question, each action being performed sequentially. Other activities may use dynamic, adaptive branching that relies on the logic of the application program 22 to inspect each response to a question and determine what the next action should be. Some activities are classified as “live.” That is, they are intended to be conducted while the handheld units 16 are communicatively linked to the computer 12.

[0027] Referring again to FIG. 1, the physical setup and software architecture of the response system 10 according to an embodiment of the invention will now be described in more detail. In this embodiment, the computer 12 is connected to the AP 12 via a USB cable. The AP 12 acts as a wireless access point on the wireless network 24. The application program 22 interacts with an operating system and service program 26 (both executing on the computer 12) to send and receive data over the USB work, through the AP 12, and over the wireless network 24 which, in this embodiment, is an IEEE 802.15.4 compliant network. To communicate with the handheld units 16, the application program 22 defines a service handler application programming interface (API) 23. The application program 22 sends hypertext transport protocol (HTTP) messages via the port along with markup language documents to the handheld units 16. The handheld units 16 respond as specified in the markup language documents, and those responses are received via the service handler API 23.

[0028] According to an embodiment of the invention, the system 10 supports two response modes: prompted and unprompted. In a prompted response mode, a question is displayed on the handheld units 16, and the audience members answer the displayed question. In an unprompted response mode, the question is not presented on the handheld units 16, but by some other method, such as a presenter reading it aloud or an audience member reading it on a handout. In an unprompted mode, the audience member still answers the question using the handheld unit 16.

[0029] In various embodiments of the invention, the application program 22 carries out activities that are paced by the presenter and activities as paced by the audience members. Since many of the examples described herein are set in the context of a classroom environment, these two types of activities will be referred to herein as “teacher-paced” and “student-paced.” It is understood, of course, that the invention is not limited to classroom use. A teacher-paced activity is one in which the class participates as a whole, and where each student responds to questions generated by the application program 22. In a teacher-paced activity, the teacher controls the pace at which questions are transmitted to the handheld units 16. For example, the students may be shown only one question at a time on their handheld units 16, and the question is shown only when specified by the teacher. Students then answer only when the teacher allows the responses to be collected. If a student fails to keep pace with the class, then the application program 22 can speed the student ahead to the current action in the activity, the next time the fallen-behind student responds in any way using his handheld unit 16. A student-paced activity is one in which each student views questions and responds at his own speed. Students can be given individual student-paced activities, or the entire class may be given the same student-paced activity. The student-paced activity may be configured so that students can navigate back and forward through questions.

[0030] An example of a teacher-paced, prompted activity will now be described with reference to FIGS. 1 and 6. First, the application program 22 transmits a mark-up language document to the plurality of handheld units 16. The mark-up language document includes instructions for presenting a sequence 60 of questions on the handheld units 16 (for a pop quiz). Each handheld unit 16 then interprets the instructions contained in the mark-up language document and, accordingly, displays the first question 52 in the sequence. Each student has the opportunity to respond to the first question 52 by making the appropriate input to the student’s handheld unit 16. When the teacher makes the appropriate input to the computer 12, the application program 22 transmits a message to the handheld units 16, that causes the handheld units to permit the students to advance to the second question 54 in the sequence 60. Note that the display containing the second question 54 includes a scroll bar 55. This process continues, with the third and fourth questions 56 and 58 being enabled in succession until the teacher ends the quiz by making the appropriate input to the computer 12.

[0031] FIG. 7 illustrates the sequence of displays shown on each handheld unit 16 for a quiz that is given in a teacher-paced, unprompted mode. The sequence of actions is similar to that described in conjunction with FIG. 6, except that the mark-up language document transmitted by the application program 22 to the handheld units 16 (FIG. 1) does not include questions but rather includes one or more answer templates, such as a True/False template 72, multiple choice template 74 and a numeric template 76. The handheld units 16 display the answer templates on their respective LCD screens 32. Furthermore, instead of the questions being presented on the LCD screen 32 of each of the handheld units 16, the teacher reads the questions aloud to the students.

[0032] According to an embodiment of the invention, the response system 10 (FIG. 1) can also be used to allow the teacher to distribute homework assignments, and to allow the students to complete and turn in the homework assignments. In such an embodiment, the application program 22 transmits one or more mark-up language documents representing a homework assignment to the handheld units 16. Outside of class, the students individually complete the homework assignment by making the appropriate inputs into their respective handheld units. FIG. 8 illustrates the sequence of displays shown on a handheld unit for an example homework assignment in a student-paced, prompted mode. The student selects "homework" from a first menu 78, then selects "Friday Quiz" from a second menu 80. In response, the handheld unit 16 displays a sequence 82 of questions (i.e., in a "prompted" mode) that the student must answer. During the next classroom session, the student makes the appropriate inputs to the handheld unit 16 to cause it to transmit her answers to the application program 22, thereby "turning in" the completed homework assignment.

[0033] The sequence of displays for another example homework assignment is shown in FIG. 9. In this example, the teacher gives handouts containing the questions to the students or displays each question on a screen that is visible to all of the students, and the mark-up language document that the handheld units 16 receive from the application program 22 only contains answer templates. Otherwise, the procedure for completing and turning in the homework assignment is the same as described in conjunction with FIG. 8.

[0034] Yet another sequence of displays for a homework assignment is shown in FIG. 10. In this example, however, the teacher gives handouts containing the questions to the students. Furthermore, the application program 22 does not transmit a mark-up language document to the handheld units 16 for answering the questions, but the students instead use a generic answer form 102 that is pre-stored in the handheld units 16. Otherwise, the procedure for answering the questions and turning in the answers is the same as described in conjunction with FIGS. 8 and 9.

[0035] According to an embodiment of the invention, the handheld unit 16 is also able to function as a calculator. FIG. 11 illustrates how this functionality appears on the LCD screen 32. The teacher may, by making the appropriate input to the computer 12 (FIG. 1), cause the application program 22 to send commands to the handheld units 16 that will disable the calculator function.

[0036] Service Handler API. As noted above, in an embodiment of the invention, the handheld units 16 communicate with the application program 22 via the wireless network 24, the AP 14, the USB cable, and the computer 12. Messages sent by the handheld units 16 to the application program 22 are received via the service handler application API 23. In one embodiment, the service handler API 23 is instantiated as an HTTP port that is provided by the application program 22. The service handler API 23 is used to host a "live" activity (i.e., an activity that requires network connectivity between the handheld units 16 and the computer 12), upload homework to the handheld units 16, upload notes to the handheld units 16, and to download homework from the handheld units 16. The service program 26 connects to the service handler API 23 when the application program 22 requests it to do so.

[0037] In an embodiment of the invention, the service program 26 provides routing services for all communications between the handheld units 16 and the application program 22 until the HTTP port for the service handler API 23 is released. Thus, messages from the handheld units 16 are first received by the service program 26, and then routed by the service program 26 to the application program 22 via the service handler API 23. In one embodiment, the messages that are sent between an application program 22 and the service program 26 are in the form of HTTP messages and XML documents, while the messages sent between the service program 26 and the handheld units 16 are in the form of compact transport protocol messages and compact markup language documents. When messages arrive from the handheld units 16, the service program converts them from compact transport protocol messages and compact markup language documents into HTTP messages and XML documents and sends the HTTP messages and XML documents to the appropriate application program 22. Conversely, the application program 22 HTTP messages and XML documents to the service program 26 that are intended for one or more of the handheld units 16, which the service program 26 converts into compact transport protocol messages and compact markup language documents. The service program 26 then transmits the compact transport protocol messages and compact markup language documents to the handheld unit or units 16.

[0038] Requests to the Service Handler API. In an embodiment of the invention, each handheld unit 16 can send messages to an application program 22 via the service program 26. When the service program 26 receives a message, from a handheld unit 16, that is intended for one of the application programs 22, the service program 26 forms a "request" and calls the service handler API 23 that corresponds to the application program 22 that is the intended recipient of the message. In general, such requests to the service handler API 23 are formed using standard HTTP GET and POST syntax of the form:

```
(GET) | (POST) + SPACE + resource_path + SPACE +
HTTP/version + CRLF
[ request-headers + CRLF ]
CRLF
[ request-body ]
```

[0039] Request headers include:

Header	Description	Example
Device-Address	MAC address (HEX) of the requesting device.	Device-Address: 32781ae3896cb366e
Request-ID	Identification of the request that should be returned in the response. Optional.	Request-ID: 101
Owner-ID	Owner identification. If an owner assignment exists for the device, the ID property is included in the request.	Owner-ID: 123456789

-continued

Header	Description	Example
User-Agent	User agent. Describes the device and version. Optional.	User-Agent: wasabi/1.0
Content-Length	Length of the request body. Omitted if no request body is present.	Content-Length: 102

[0040] Consider the example services URL used to get a list of files that an application program **22** has assigned for download to a handheld unit **16**:

```

/service_url/GetFileList
The HTTP request might look like:
GET /service_url/GetFileList HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0

```

[0041] When an application program **22** responds to a request from the service program **26**, it does so using HTTP syntax of the form:

```

HTTP/version + SPACE + status_code + SPACE +
response_status + CRLF
[ response-headers + CRLF ]
CRLF
[ response-body ]

```

[0042] Response headers include:

Header	Description	Example
Device-Address	MAC address (HEX) of the requesting device.	Device-Address: 32781ae3896cb366e
Request-URL	URL of request the response is intended for.	Request-URL: /ServiceHandler/64/Authenticate
Request-ID	Identification of the request that is returned if included in the original response. Optional.	Request-ID: 101
Server	Identifies the application program providing the response. Optional.	Server:AccelTest/2.0
Content-Length	Length of the response body.	Content-Length: 102

[0043] The response-body is enclosed in a <data> element and includes a <status> element whose contents provide a result code and optional message for the caller. This result code and message differs from the HTTP status code and response status, which relate to the success of locating and calling the service.

[0044] Example:

```

HTTP/1.1 200 OK
Device-Address: 32781ae3896cb366e
Request-URL: /ServiceHandler/64/Authenticate
Request-ID: 101
Server: AccelTest/1.0
Content-Length: 54
<data>
  <status code="result code">
    <message>some text here</message>
  </status>
</data>

```

[0045] The <authenticate> element is returned when an application program **22** requires the user to identify themselves before a service can be successfully completed. On receiving this result, the handheld unit **16** will provide a general input field to input an alphanumeric id, which will be returned to service handler API of the application program **22**.

[0046] Example:

```

<data>
  <status code="401">Unauthorized</status>
  <authenticate>
    <application id="5899128ca5637be36"
      name="AccelTest" />
    <prompt>Enter your student ID:</prompt>
  </authenticate>
</data>

```

[0047] Attributes

[0048] admin=[true|false]

[0049] Authenticate administrator PIN instead of user key. False is default.

[0050] no_keychain=[true|false]

[0051] If true, the device should not store the user's credentials for this authentication. False is default.

[0052] Content

[0053] application: id

[0054] ID of the application requesting authentication.

[0055] application: name

[0056] Name of the application requesting authentication.

[0057] <prompt>

[0058] A required element specifying the text prompt to display for login identification.

[0059] The following are examples of requests that are sent by the service program **26** to the application programs **22** via the service handler API **23** (in response to a handheld unit **16** invoking the request), and the responses to sent by the application programs **22** to the service program **26** (which are subsequently converted and transmitted to the handheld unit **16**).

[0060] Authenticate request. The Authenticate request is invoked by a handheld unit 16 (via the service program 26) that is responding to a request from an application program 22 to identify the user.

[0061] Format of an Authenticate Request

[0062] POST/service_url/Authenticate HTTP/1.1

Parameter	Validation	Value
application/id	Not Null	The id of the application for which the device is requesting authentication.
key	Required for user authentication. Optional for admin authentication.	Authentication key for the user or administrator.
key/admin	Optional	True or false. True indicates the authentication is for the administrator PIN
key/validated	Optional. Required with admin attribute	True or false. True indicates the admin PIN the device supplied is valid.

[0063] Example Request:

```
POST /ServiceHandler/64/Authenticate HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
Content-Length: 58
<data>
  <credentials>
    <key admin="true" validated="true">1234</key>
    <application id="5899128ca5637be36" />
  </credentials>
</data>
```

[0064] Response to an Authenticate request

[0065] Example Response #1:

```
HTTP/1.1 200 OK
Device-Address: 32781ae3896cb366e
Request-URL: /ServiceHandler/64/Authenticate
Request-ID: 101
Content-Length: 33
<data>
  <status code="200">OK</status>
</data>
```

[0066] Example Response #2:

```
HTTP/1.1 200 OK
Device-Address: 32781ae3896cb366e
Request-URL: /ServiceHandler/64/Authenticate
Request-ID: 101
Content-Length: 78
<data>
  <status code="403">You have been locked out. See your
  teacher.</status>
</data>
```

[0067] Status Codes for Service Program 26:

Code	Description
200	OK
401	Unauthorized
403	Forbidden
500	Internal server failure

[0068] HTTP Status Codes:

Code	Description
200	OK
400	Bad request
404	Not found
408	Request timeout
500	Internal server failure
503	Service unavailable

[0069] GetLiveActivity request. The GetLiveActivity request is used to obtain the current live activity.

[0070] Format of a GetLiveActivity Request

[0071] Post/service_url/GetLiveActivity HTTP/1.1

Parameter	Validation	Value
activity: id	Optional	ID of the activity the responses are intended for.
application: id	Optional element; Required if response data	GUID identifying the application server the response is intended for.
activity: submitted	is included. Optional	Indicates activity has been submitted/completed by the user. Value is "true" or "false". "false" is the default.
response: actionid	Optional	Number ID of the action the response is intended for.
response: question	Optional	Number of question for which the response is intended
response: start_question	Optional	Starting question number for the range of responses being submitted.
response: end_question	Optional	Ending question number for the range of responses being submitted.
response:text()	Optional	Text of the response. For multiple question submission, responses are comma delimited.
navigation: question	Optional. Mutually exclusive with back and forward attributes.	Next question number the user has requested to view.

-continued

Parameter	Validation	Value
navigation: back	Optional. Mutually exclusive with question and forward attributes.	User requested to navigate backwards if value is "true"
navigation: forward	Optional. Mutually exclusive with question and back attributes.	User requested to navigate backwards if value is "true"
credentials	Optional. Included for inline authentication.	

[0072] Example Request 1: Simple request.

```
POST /ServiceHandler/64/GetLiveActivity HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
```

[0073] Example Request 2: Simple question response submission.

```
POST /ServiceHandler/64/GetLiveActivity HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
Content-Length: 75
<data>
  <activity id="43920afe128ca82a1">
    <application id="5899128ca5637be36" />
    <response actionid="4"
      question="1">b</response>
  </activity>
</data>
```

[0074] Example Request 3: Response submission with navigation option.

```
POST /ServiceHandler/64/GetLiveActivity HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
Content-Length: 97
<data>
  <activity id="43920afe128ca82a1">
    <application id="5899128ca5637be36" />
    <response actionid="5" question="4">43.1</response>
    <navigation question="1"/>
  </activity>
</data>
```

[0075] Example Request 4: Repeater question response submission.

```
POST /ServiceHandler/64/GetLiveActivity HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
Content-Length: 142
```

-continued

```
<data>
  <activity id="43920afe128ca82a1">
    <application id="5899128ca5637be36" />
    <response actionid="4" start_question="1"
      end_question="15">b,a,,ab,c,e,d,abc,d,,,a,d</response>
  </activity>
</data>
```

[0076] Example Request 5: Response submission with submit (completed).

```
POST /ServiceHandler/64/GetLiveActivity HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
Content-Length: 97
<data>
  <activity id="43920afe128ca82a1" submitted="true" >
    <application id="5899128ca5637be36" />
    <response actionid="5"/>
  </activity>
</data>
```

[0077] Example Request 6: Request with credentials

```
POST /ServiceHandler/64/GetLiveActivity HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
Content-Length: 97
<data>
  <credentials>
    <key admin="true" validated="true">1234</key>
    <application id="5899128ca5637be36" />
  </credentials>
</data>
```

[0078] Response to a GetLiveActivity request

```
HTTP/1.1 200 OK
Device-Address: 32781ae3896cb366e
Request-URL: /ServiceHandler/64/GetLiveActivity
Request-ID: 101
Content-Length: 390
<data>
  <status code="200" />
  activity xml.
</data>
```

[0079] Status Codes for service program 26:

Code	Description
200	OK
401	Unauthorized
403	Forbidden
500	Internal server failure

[0080] HTTP Status Codes:

Code	Description
200	OK
400	Bad request
404	Not found
408	Request timeout
500	Internal server failure
503	Service unavailable

[0081] GetFileList request. The GetFileList request is invoked when a handheld unit **16** requests an updated list of files it should download. Typically, these files include homework and notes.

[0082] Format of a GetFileList Request**[0083]** GET/service_url/GetFileList HTTP/1.1

Parameter	Validation	Value
None.		

[0084] Example Request 1: Simple Request

```
GET /Wasabi/GetFileList HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
```

[0085] Example Request 2: Request with credentials

```
POST /ServiceHandler/64/GetFileList HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
Content-Length: 97
<data>
  <credentials>
    <key admin="true" validated="true">1234</key>
    <application id="5899128ca5637be36"> />
  </credentials>
</data>
```

[0086] Response to a GetFileList request

```
HTTP/1.1 200 OK
Device-Address: 32781ae3896cb366e
Request-ID: 101
Content-Length: 145
<data>
  <status code="200"> />
  <filelist>
    <application id="3d8920a182ef4829"> />
    <files>
      <homework id="ab7291ef3910a121">
        <title>Take home quiz</title>
      </homework>
```

-continued

```
<homework id="672fa18bc6371a36">
  <title>Ch1.1 Worksheet</title>
</homework>
<note id="42819ae637bc8163"> />
</files>
</filelist>
</data>
```

[0087] Status Codes for service program **26**:

Code	Description
200	OK
401	Unauthorized
403	Forbidden
500	Internal server failure

[0088] HTTP Status Codes:

Code	Description
200	OK
400	Bad request
404	Not found
408	Request timeout
500	Internal server failure
503	Service unavailable

[0089] GetFile request. The GetFile request is invoked when a handheld unit **16** requests a file to download.

[0090] Format of a GetFile Request**[0091]** POST/service_url/GetFile HTTP/1.1

Parameter	Validation	Value
application/id	Not Null	The guid identifying the application.
file/id	Not Null	The guid identifying the homework file.

[0092] Example Request:

```
POST /Wasabi/GetFile HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
Content-Length: 64
<data>
  <file id="ab7291ef3910a121">
    <application id="3d8920a182ef4829"> />
  </file>
</data>
```

[0093] Response to a GetFile request

```

HTTP/1.1 200 OK
Device-Address: 32781ae3896cb366e
Request-ID: 101
Content-Length: 612
<data>
  <status code="200" />
  <file>
    homework or note markup
  </file>

```

[0094] </data>**[0095]** Status Codes for service program 26:

Code	Description
200	OK
401	Unauthorized
500	Internal server failure
516	File not found

[0096] HTTP Status Codes:

Code	Description
200	OK
400	Bad request
404	Not found
408	Request timeout
500	Internal server failure
503	Service unavailable

[0097] ConfirmGetFile request. The ConfirmGetFile request is invoked to confirm that the handheld unit **16** received the specified file.

[0098] Format of a ConfirmGetFile Request**[0099]** POST/service_url/ConfirmGetFile HTTP/1.1

Parameter	Validation	Value
application/id	Not Null	The guid identifying the application.
filetype/id	Not Null	The ID identifying the file. Filetype is either note or homework.

[0100] Example Request:

```

POST /Wasabi/ConfirmGetFile HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
Content-Length: 64
<data>
  <file id="ab7291ef3910a121">
    <application id="3d8920a182ef4829" />
  </file>
</data>

```

[0101] Response to a ConfirmGetFile request

```

HTTP/1.1 200 OK
Device-Address: 32781ae3896cb366e
Request-ID: 101
Content-Length: 33
<data>
  <status code="200" />
</data>

```

[0102] Status Codes for service program 26:

Code	Description
200	OK
401	Unauthorized
500	Internal server failure
516	File not found

[0103] HTTP Status Codes:

Code	Description
200	OK
400	Bad request
404	Not found
408	Request timeout
500	Internal server failure
503	Service unavailable

[0104] PutFile request. The PutFile request is invoked when a handheld unit **16** requests to upload a file to the application program **22**.

[0105] Format of a PutFile Request**[0106]** POST/service_uri/Putfile HTTP/1.1

Parameter	Validation	Value
application/id	Not Null	The guid identifying the application.
homework/id	Not Null	The ID identifying the file.
homework file data	Not Null	File data.

[0107] Example Request:

```

POST /Wasabi/Putfile HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
Content-Length: 143
<data>
  <file id="ab7291ef3910a121">
    <application id="3d8920a182ef4829" />
    homework markup
  </file>

```

[0108] </data>

[0109] Response to a PutFile request

```

HTTP/1.1 200 OK
Device-Address: 32781ae3896cb366e
Request-ID: 101
Content-Length: 33
<data>
  <status code="200" />
</data>

```

[0110] Status Codes for service program 26:

Code	Description
200	OK
401	Unauthorized
500	Internal server failure
516	File not found
517	Disk full

[0111] HTTP Status Codes:

Code	Description
200	OK
400	Bad request
404	Not found
408	Request timeout
500	Internal server failure
503	Service unavailable
516	File not found
517	Disk full

[0112] According to an embodiment of the invention, when a “live” activity is in progress, the application program 22 acts as a web server providing one or more compact mark-up language documents representing the live activity to the handheld units 16, which act as a browsers. Each handheld unit displays information on the LCD screen 32 as specified by the compact mark-up language document. The compact mark-up language document also dictates how the user may navigate, input responses, and what prompts the user receives. The application program 22 provides this live activity “page” to the service program 26 when the service program 26 sends a GetLiveActivity request to the application program 22 (in response to a corresponding request from a handheld unit 16). Below is a sample request-response loop demonstrating a simple live activity session:

[0113] Request from service program 26 (initiated by student’s handheld unit 16):

```

POST /GetLiveActivity HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0

```

[0114] Response from application program 22:

```

HTTP/1.1 200 OK
Content-Length: 789
<data>
  <status code="200" />
  <activity id="43920afe128ca82a1">
    <application id="1234567890123456"/>
    <title>Chapter 1 Quiz</title>
    <actions>
      <respond id="1">
        <prompt>What is the capital of Wisconsin?</prompt>
        <question type="abcd" number="1" total_questions="20"
          multiple_select="false">
          <distractors>
            <distractor>Mosinee</distractor>
            <distractor>Madison</distractor>
            <distractor>Milwaukee</distractor>
            <distractor>Wisconsin Rapids</distractor>
          </distractors>
        </question>
      </respond>
    </actions>
  </activity>
</data>

```

[0115] The service program 26 converts the XML into a compact markup language document and transmits it to the handheld unit 16. The handheld unit 16 displays the question to the student. When the student responds to the question, the handheld unit 16 will again initiate a GetLiveActivity request, this time providing data in the query parameters which identify the activity, the question, and the user’s response. The following example shows the request that the gets passed from the service program 26 to an application program 22 in response to the service program 26 receiving a corresponding request from the handheld unit 16:

```

POST /Wasabi/GetLiveActivity HTTP/1.1
Device-Address: 32781ae3896cb366e
Request-ID: 101
User-Agent: Wasabi/1.0
Content-Length: 64
<data>
  <status code="200" />
  <activity id="43920afe128ca82a1">
    <application id="1234567890123456"/>
    <response actionid="1" question="1">b</response>
  </activity>
</data>

```

The application program 22’s response to this request can be dynamic based on the activity, the action, and the response to the question.

[0116] The following sections discuss the elements of an activity document and the different action types that can be used by the application program 22 to create the live and homework style activities. Although the example activity documents are implemented in XML, any markup language may be used.

[0117] As previously discussed, the activity document is the content that the application program 22 provides via the service handler API 23 when it receives a GetLiveActivity request. An activity document includes a single <activity> element that itself can contain zero or more action elements.

Action elements direct the user experience for the handheld unit **16**. The following is an example of an activity document that uses the <activity> element:

```
<activity id="43920afe128ca82a1">
  <application id="1234567890123456"/>
  <environment_settings>
    <calculator enabled="false" />
  </environment_settings>
  <title>Chapter 1 Quiz</title>
  <actions>
    <respond id="1">
      <prompt>What is the capital of Wisconsin?</prompt>
      <question type="abcd" number="1">
        <distractors>
          <distractor>Mosinee</distractor>
          <distractor>Madison</distractor>
          <distractor>Milwaukee</distractor>
          <distractor>Wisconsin Rapids</distractor>
        </distractors>
      </question>
    </respond>
    <respond id="2">
      <prompt>What is our main export?</prompt>
      <question type="abcd" number="2">
        <distractors>
          <distractor>Beef</distractor>
          <distractor>Cheese</distractor>
          <distractor>Cranberries</distractor>
          <distractor>Apples</distractor>
        </distractors>
      </question>
    </respond>
    <respond id="3">
      <prompt>Who is the governor of Wisconsin?</prompt>
      <question type="abcd" number="3">
        <distractors>
          <distractor>Tommy Thompson</distractor>
          <distractor>Bret Farve</distractor>
          <distractor>Jim Doyle</distractor>
          <distractor>George Bush</distractor>
        </distractors>
      </question>
    </respond>
    <respond id="4">
      <prompt>The Packers rule!</prompt>
      <question type="tf" number="4" />
    </respond>
    <respond id="5">
      <prompt>What is the population of Madison, WI?</prompt>
      <question type="numeric" number="5" />
    </respond>
    <submit>
      <prompt>Are you ready to turn in your assignment?</prompt>
    </submit>
  </actions>
</activity>
```

In above example, the activity includes 5 actions to gather responses from the student and 1 action to handle submission of the work when the student reaches the end of the quiz.

[0118] The following sections describe the elements of an activity document according to an embodiment of the invention, including optional elements for controlling device settings and user navigation. The following sections also describe the format of the corresponding compact markup language documents that are transmitted to the handheld units **16**.

[0119] The <activity> element. The <activity> element forms the root of the activity document.

[0120] Example:

```
<activity id="43920afe128ca82a1">
  content elements
</activity>
```

If the application program **22** has no live activity to provide to the requesting device, the null activity should be returned using the empty element form <activity />.

[0121] Attributes

[0122] id

[0123] An optional attribute that the application program **22** can supply to uniquely identify the activity.

[0124] submitted="[true|false]"

[0125] An optional attribute used in live activities to indicate the activity is submitted as complete. Default is false.

[0126] Content

[0127] <application>

[0128] A required element that identifies the application the activity was served from.

[0129] <environment_settings>

[0130] An optional element that specifies what device settings are in effect during the activity. Settings allow utilities such as the calculator to be turned on or off as dictated by the teacher.

[0131] <title>

[0132] A required element that provides a human readable name for the activity. Examples include: Chapter 4 Quiz, Customer Survey, Math Assessment.

[0133] <actions>

[0134] An optional element which includes sub elements defining the action or actions that the handheld unit **16** should perform for the activity.

[0135] <navigation>

[0136] An optional element defines the back, forward, and indexed navigation the user can perform for the activity.

[0137] Compact Markup Language

Item	Description	Usage
act	Activity element	
act\i	Activity id attribute	
Act\sub	Submitted attribute	\sub0 = do not submit \sub1 = submit as completed

[0138] Example

[0139] {act\i 1234567890123456{\t Chapter 1 Quiz}}

[0140] The <application> element. The <application> element identifies the application program that served the activity.

[0141] Example:

```
<activity id="43920ate128ca82a1">
  <application id="1234567890123456"/>
  content elements
</activity>
```

[0142] Attributes

[0143] id

[0144] A required attribute that identifies the application program 22. The value is required to be a unique GUID in 32 character ascii-hex.

[0145] name

[0146] An optional attribute that identifies the application name that served the activity.

[0147] Compact Markup Language

Item	Description	Usage
app	Application element	
app\i	Application id attribute	Unique GUID; required
app\n	Application name	Optional

[0148] Example

[0149] {app\i 1234567890123456\n AccelTest}

[0150] The <environment_settings> element. The <environment_settings> element contains sub elements that specify the device environment settings for the activity.

[0151] Example:

```
<environment_settings >
  content elements
</environment_settings>
```

[0152] Attributes

[0153] None.

[0154] Content

[0155] <calculator>

[0156] An optional element that specifies whether the calculator is to be enabled or disabled during the activity.

[0157] Compact Markup Language

Item	Description	Usage
evs	Environment settings element	

[0158] Example

[0159] {act\i 1234567890123456{evs{calc\e}}}

[0160] The <calculator> element. The <calculator> element is a sub element of the <environment_settings> element and specifies whether the calculator is available during an activity. If this element is not provided, the calculator is enabled by default.

[0161] Example:

[0162] <calculator enabled="true"/>

[0163] Attributes

[0164] enabled="[true|false]"

[0165] A required attribute specifying the enabled state of the calculator utility on the Wasabi. Default value is true.

[0166] Content

[0167] None.

[0168] Compact Markup Language

Item	Description	Usage
calc\e	calculator element Enabled attribute	If present, calculator is enabled.

[0169] Example {act\i 1234567890123456{evs{calc\e}}}

[0170] The <title> element. The <title> element is a sub element of the <activity> element and specifies the title of the activity.

[0171] Example:

[0172] <title>Chapter 4 Quiz</title>

[0173] Attributes

[0174] None.

[0175] Content

[0176] Text.

[0177] Compact Markup Language

Item	Description	Usage
t	title element	

[0178] Example

[0179] {act\i 1234567890123456{t\ Chapter 1 Quiz}}

[0180] The <navigation> element. The <navigation> element is a sub element of the <activity> element and specifies settings for user navigation using the keypad and menu system of the handheld unit 16.

[0181] Example:

```
<navigation back="false" forward="true" first_question="1"
last_question="20" />
```

[0182] Attributes

[0183] back="[true|false]"

[0184] An optional attribute specifying the user can navigate backwards in an activity. Defaults to false.

[0185] forward="[true|false]"

[0186] An optional attribute specifying the user can navigate forwards in an activity. Defaults to false.

[0187] first_question

[0188] An optional attribute specifying the number of the first question the user can navigate to using the indexed navigation UI of the handheld unit 16. If omitted or 0, indexed navigation is disabled. Value is limited from 0 to 999.

[0189] last_question

[0190] An optional attribute specifying the number of the last question the user can navigate to using the indexed navigation UI of the handheld unit 16. If omitted or 0, indexed navigation is disabled. Value is limited from 0 to 999.

[0191] Content

[0192] None.

[0193] Compact Markup Language

Item	Description	Usage
nv	navigation element	
\b	back attribute	If present, backward navigation is supported.
\f	forward attribute	If present, forward navigation is supported.
\s	First question attribute	Specifies starting question number
\e	Last question attribute	Specified ending question number

[0194] Example

[0195] {nv\b\s1\e25}

[0196] The <actions> element. The <actions> element is a sub element of the <activity> element and is used to specify the actions for the activity.

[0197] Attributes

[0198] None.

[0199] Content

[0200] Allowed elements include: <respond>, <take-note>, <message>, <submit>, <continue>

[0201] The <respond> element. The <respond> element is an action sub element of the <actions> element used to direct the handheld unit 16 to display a question prompt and then collect an input response from the student.

[0202] Example:

```
<respond id="4" >
  <prompt>What is the population of Madison, WI?</prompt>
  <question type="numeric" number="5" />
  <response>345000</response>
</respond >
```

[0203] Attributes

[0204] id

[0205] An optional attribute used to identify the action. The id of the action is passed back to the application program 22 on the next call to GetLiveActivity. The id may be used to help manage state and session in the application program 22 in a complex sequence activity. The value of this attribute must be a numeric from 0 to 9999.

[0206] Content

[0207] <prompt>

[0208] An optional element specifying the text prompt to display.

[0209] <question>

[0210] An optional element specifying the type of response desired.

[0211] <response>

[0212] An optional element specifying a response that was recorded previously.

[0213] Compact Markup Language

Item	Description	Usage
r	Respond element	
\i	ID attribute	Action id
rs	Response element	

[0214] Example

[0215] {r\i2{p\ Saturn is the _____ planet from the sun.}{q\ t n}}

[0216] The <message> element. The <message> element is an action sub element of the <actions> element used to direct the handheld unit 16 to display a prompt and then wait for a keypad press from the student.

[0217] Example:

```
<message id="5">
  <prompt>Open your book to page 129 to answer the next 5 questions.
  Press any key to continue.</prompt>
  <keys>any</keys>
</message>
```

[0218] Attributes

[0219] id

[0220] An optional attribute used to identify the action. The id of the action is passed back to the application program 22 on the next call to GetLiveActivity. The id may be used to help manage state and session in the application program 22 in a complex sequence activity. The value of this attribute is a numeric from 0 to 9999.

[0221] Content

[0222] <prompt>

[0223] A required element specifying the message prompt to display.

[0224] <keys>

[0225] An optional element specifying keypad buttons that will dismiss the message and advance the user to the next activity action.

[0226] Compact Markup Language

Item	Description	Usage
m i	Message element ID attribute	Action id

[0227] Example

[0228] {m\16{p\ Do you wish to continue?}{k\ y,n}}

[0229] The <takenote> element. The <takenote> element is an action sub element of the <actions> element used to direct the handheld unit **16** to display a note and then wait for the user to accept or reject the note.

[0230] Example:

```
<takenote id="7">
  <note id="3875a28cd6185eda" locked="false">
    <title>Score Results</title>
    <from>Mrs. Smith</from>
    <date>20050612</date>
    <body>Congratulations. You scored 85% on the Chapter 1 quiz.
Be sure to read the next chapter, sections 1–4 for tomorrow's
class.</body>
  </note>
</takenote>
```

[0231] Attributes**[0232]** id

[0233] An optional attribute used to identify the action. The id of the action is passed back to the application program **22** on the next call to GetLiveActivity. The id may be used to help manage state and session in the application program **22** in a complex sequence activity. The value of this attribute is a numeric from 0 to 9999.

[0234] Content**[0235]** <note>

[0236] A standard note file element.

[0237] Compact Markup Language

Item	Description	Usage
tn i	takenote element ID attribute	Action id
no	Note element	
no i	Note ID attribute	Required
no l	Note locked attribute	If present, not is locked.
t	Title element	Optional
fr	From element	Optional
dt	Date element	Optional
bd	Body element	Required

[0238] Example

```
{tn\i8{no\i 1234l{t\ Score Results}{fr\ Mrs. Smith}{dt\
20050015}{bd\
Congratulations. You scored 85% on the quiz. Be sure to read the next
chapter, sections 1–4 for tomorrow's class.}}}
```

[0239] The <submit> element. The <submit> element is an action sub element of the <actions> element used to direct the handheld unit **16** to display a prompt asking the user if the activity is completed and ready to be turned in. This element is typically used when the student has responded to the last question in a live student paced activity or a homework activity. The handheld unit **16** will display the prompt and expect to collect a Yes or No response.

[0240] Examples:

```
<submit>
  <prompt>Are you ready to turn in your homework?</prompt>
</submit>
<submit>
  <prompt>Are you ready to turn in your quiz?</prompt>
</submit>
<submit />
```

[0241] Content**[0242]** <prompt>

[0243] An optional element specifying the text prompt to display. If omitted, a standard prompt on the device is used.

[0244] Compact Markup Language

Item	Description	Usage
sb p	submit element prompt element	

[0245] Example

[0246] {sb{p\ Are you ready to turn in your assignment?}}

[0247] The <continue> element. The <continue> element is an action sub element of the <actions> element used to direct the handheld unit **16** to allow the user to continue with the next action in an activity sequence.

[0248] Example:

[0249] <continue />

[0250] Attributes

[0251] None.

[0252] Content

[0253] None.

[0254] Compact Markup Language

Item	Description	Usage
con	Continue element	

[0255] Example

[0256] {act\i 1234{con}}

[0257] The <prompt> element. The <prompt> element is a sub element of the <respond>, <message>, and <submit> elements and specifies the text that will be displayed to the user of the handheld unit 16.

[0258] Examples:

```
<prompt>What is the capital of Wisconsin?</prompt>
<prompt>Turn to page 140 in your Math textbook and answer
the following multiple choice questions.</prompt>
<prompt>Have you finished the quiz?</prompt>
```

[0259] Attributes

[0260] None.

[0261] Content

[0262] Text.

[0263] Compact Markup Language

Item	Description	Usage
P	prompt element	

[0264] Example

[0265] {r\i2{p\ Saturn is the _____ planet from the sun.}{q\t n}}

[0266] The <keys> element. The <keys> element is a sub element of the <message> action element and is used to specify what keypad buttons will advance the user to the next activity action. If omitted, then any key will dismiss the message.

[0267] Examples:

```
<keys>any</keys>
<keys>ent</keys>
<keys>a,b,c,d,e,tru,fal,ent</keys>
```

[0268] Attributes

[0269] None.

[0270] Content

[0271] A comma delimited string of keypad buttons. Valid values are listed in the table below.

Key Description	Key Code
A	a
B	b
C	c
D	d
E	e

-continued

Key Description	Key Code
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
Enter	ent
Clear	clr
Menu	mmu
Select	sel
Up Arrow	up
Down Arrow	dwn
Period/Decimal Point	.
Plus/Minus	+−
True	tru
False	fal
Yes	yes
No	no
Question Mark	?

[0272] Compact Markup Language

Item	Description	Usage
k	Keys element	Valid values include are listed in table 3-1.

[0273] Example

[0274] {m\i6{p\ Do you wish to continue?}{k\ yes,no}}

[0275] The <question> element. The <question> element is a sub element of the <respond> action element and is used to specify what keypad buttons are valid for collecting a response from the user and other properties specific to the type of response being requested.

[0276] Example 1: A multiple choice question with one correct answer.

```
<question type="abcd" number="2" total_questions="4"
multiple_select="false">
  <distractors>
    <distractor>Mosinee</distractor>
    <distractor>Madison</distractor>
    <distractor>Milwaukee</distractor>
    <distractor>Wisconsin Rapids</distractor>
  </distractors>
</question>
```

[0277] Example 2: A true/false question.

[0278] <question type="tf" number="2" total_questions="4"/>

[0279] Example 3: A numeric question.

[0280] <question type="numeric" number="2" total_questions="4"/>

[0281] Example 4: A multiple choice repeater question. Used for scan card style activities. `<question type="abcd" start_number="1" end_number="50" total_questions="50" multiple_select="true" alternate_distractors="true"/>`

[0282] Attributes

[0283] type

[0284] An optional attribute specifying the allowed response type. Valid values include: ab, abc, abcd, abcde, fg, fgh, fghj, fghjk, tf, yn, numeric, free, none.

[0285] number

[0286] An optional attribute specifying the question number. This number is sent back to the application program 22 when a response is submitted by the user. Value is from 0 to 999.

[0287] start_number

[0288] An optional attribute specifying the starting question number of a repeater question. Not valid to use with number attribute. Requires use of end_number attribute.

[0289] end_number

[0290] An optional attribute specifying the ending question number of a repeater question. Not valid to use with number attribute. Requires use of start_number attribute.

[0291] total_questions

[0292] An optional attribute specifying the total number of questions in the activity. Together with the number attribute, the handheld unit 16 will use this information when displaying the question on the screen to the user as shown, for example, in FIG. 8. Note that on the upper right of the screen (reference number 82) shows the total number of questions (e.g., "2 of 4").

[0293] multiple_select=[true|false]

[0294] An optional attribute specifying the user can enter more than one answer for a question of type: ab, abc, abcd, abcde.

[0295] Default value is false.

[0296] alternate_distractors=[true|false]

[0297] An optional attribute for repeater questions that specifies the use of alternating distractors labels for even numbered questions. For example, a question of type "abcde" will use "fghjk" labels for even questions and "abcde" labels for odd numbered questions. Default value is false. Not valid to use with number attribute. Requires use of start_number and end_number attributes.

[0298] Content

[0299] <distractors>

[0300] An optional element valid only when the question type is set to one of the following values: ab, abc, abcd, abcde. Distractors are labels defined by the application program 22 to display to the user when asking a multiple-choice question.

[0301] Compact Markup Language

Item	Description	Usage
q	question element	
\t	type attribute	Required. Valid values include: b, c, d, e, g, h, j, k for multiple choice, n for numeric, y for yes/no, t for true/false, and fr for free response.
\i	number attribute	
\si	start_number attribute	
\ei	end_number attribute	
\n	total_questions attribute	
\ms	multiple_select attribute	If present, multiple select is enabled.
\a	alternate_distractors attribute	If present, alternate distractor labels are enabled.

[0302] Example 1

[0303] `{r\i2{p\ Saturn is the _____ planet from the sun.}{q\i2\n10\t n}}`

[0304] Example 2. A 50 question multiple choice repeater.

[0305] `{r\i2{q\si1\ei50\n10\t e\ms\a}}`

[0306] The <distractors> element. The <distractors> element is a sub element of the <question> element and is used to specify the multiple choice distractor elements.

[0307] Example 1: A multiple choice question with one correct answer.

```
<question type="abcd" number="2" total_questions="4">
  <distractors>
    <distractor>Mosinee</distractor>
    <distractor>Madison</distractor>
    <distractor>Milwaukee</distractor>
    <distractor>Wisconsin Rapids</distractor>
  </distractors>
</question>
```

[0308] Attributes

[0309] None.

[0310] Content

[0311] <distractor>

[0312] A required element which specifies the text of a multiple-choice answer.

[0313] The <distractor> element. The <distractor> element is a sub element of the <distractors> element and is used to specify the text of a multiple choice answer.

[0314] Example:

[0315] `<distractor>Mosinee</distractor>`

[0316] Attributes

[0317] None.

[0318] Content

[0319] Text.

[0320] Compact Markup Language

Item	Description	Usage
d	distractor element	

[0321] Example

```
{r;2{p} Which planet is the sixth planet from the Sun?}{q;i2\n10\t d{d\
Mars}{\d Jupiter}{\d Saturn}{\d Venus}}}
```

[0322] Homework is a special type of student-paced activity in which the activity is downloaded to the handheld unit 16 and completed offline. A live connection to the wireless network 24 and the application program 22 is not required for a student to engage in a homework activity. Since the activity is completed offline, the application program 22 generally does not intervene to control the pace and flow of homework activity. Thus, homework activities are generally static activities. Homework is not classified as “live” as are teacher and student-paced activities.

[0323] Completing a homework activity involves the student responding to the questions in the activity and marking the activity complete. To allow the application program 22 to collect homework, the student connects to the wireless network 24 using her handheld unit 16 and updates the application program 22. During the update, answers to the homework activity questions are uploaded to the application program 22's service handler API 23.

[0324] In an embodiment of the invention, a handheld unit 16 homework activity can be “locked” when it is created by the application program 22. A locked homework activity cannot be deleted explicitly by the user of the handheld unit 16. It can only be deleted by the teacher using the application program 22 or by submitting it as completed. Locked homework activities are typically used when a teacher wants to make sure a student does not delete the assignment (either mistakenly or purposefully).

[0325] A homework document is a markup language document, such as an XML document. The application program 22 may provide a homework document to a handheld unit 16 via the service handler API 23 and the service program 26 when the handheld unit 16 submits a GetFile request. The homework document is similar to the activity document, with a few small differences. A homework document includes at least one <homework> element. The <homework> element has one required attribute, “id” and one optional attribute, “locked.” The “id” attribute identifies the homework activity. Each homework activity the application program 22 creates should be assigned a unique identifier. This identifier is used to track the note file from creation to deletion. A GUID is typically used, although any id that is unique to the handheld unit 16 user is sufficient. The “locked” attribute can have a value of true or false, and indicates whether or not the homework activity is locked. Homework activities that are “locked” cannot be deleted by the handheld unit 16 user. The default value for the locked attribute is false. The <homework> element includes the optional elements <sender>, <assign_date> and <due_date>. The <sender> element specifies the name of the person that assigned the activity. The <assign_date> element specifies

the date the activity was assigned. The date format is CCYYMMDD. The <due_date> element specifies the date the activity is due. The date format is CCYYMMDD.

[0326] The following is an example of a homework document

```
<homework id="3875a28cd6185eda" locked="false">
  <sender>Mrs. Smith</sender>
  <assign_date>20050612</assign_date>
  <due_date>20050616</due_date>
  standard activity content goes here
</homework>
```

[0327] Attributes**[0328]** id

[0329] A required attribute used to identify the homework activity. Each homework activity the application program 22 creates should be assigned a unique identifier. This id is used to track the homework from creation to deletion. A GUID is typically used although any id that is unique to the Wasabi user is sufficient.

[0330] locked=[true|false]

[0331] An optional attribute indicating the lock state of the homework activity. Homework activities that are “locked” cannot be deleted by the user of the handheld unit 16. Default value is false.

[0332] Content**[0333]** <application>

[0334] A required element that identifies the application the homework was served from.

[0335] <from>

[0336] An optional text element specifying the name of the person that assigned the activity.

[0337] <assign_date>

[0338] An optional element specifying the date the activity was assigned. Date format is CCYYMMDD.

[0339] <due_date>

[0340] An optional element specifying the date the activity is due. Date format is CCYYMMDD.

[0341] Homework response document. A completed homework document is a markup language document that the application program 22 receives via the service handler API 23 when PutFile is invoked by a handheld unit 16 to submit completed homework. The document defines the responses entered by the student for the homework activity.

[0342] Example #1:

```
<homework submitted="true">
  <responses>
    <response actionid="4" question="1">b</response>
  </responses>
</homework>
```

[0343] Attributes**[0344]** submitted**[0345]** A flag indicating the student marked the homework activity as completed. If false, the student has not yet completed the homework.**[0346]** Content**[0347]** <response>**[0348]** A element identifying a response to an activity action.**[0349]** actionid**[0350]** ID of the action**[0351]** question**[0352]** An optional attribute identifying the question number.**[0353]** If the application program 22 supports homework, then it may also provide features to create homework. If the application 22 supports “locked” homework, then the application program 22 may also include features to delete locked homework that is never turned in by the student.**[0354]** When a handheld unit 16 submits a GetFileList request via the service program 26 and the service handler API 23, the application program 22’s response should include in the list any homework it wants downloaded to the requesting handheld unit 16. The homework files themselves are then provided by the application program 22 when the handheld unit 16 submits a GetFile request via the service handler program 26.**[0355]** When a homework activity is successfully downloaded to the handheld unit 16, the handheld unit 16 will submit a ConfirmGetFile request to the application program 22 using the id of the homework activity the application program 22 assigned.**[0356]** When the student completes the homework activity, she will use the handheld unit 16 user interface to mark it complete and ready for submission. Homework that is completed will be uploaded to the application program 22 using the PutFile request.**[0357]** Typically submitted homework will be deleted from the user’s file list after the homework file is successfully uploaded with the PutFile request. After each PutFile invocation, the handheld unit 16 will submit a GetFileList request to confirm the file was received and removed from the homework list. If the handheld unit 16 finds the file has been removed from the list, it will delete it from its own internal memory.**[0358]** If the handheld unit 16 finds, in its internal memory, homework that came from the application program 22 but is no longer reported in the application program 22’s file list, then the handheld unit 16 will attempt to delete the file or files for the homework from its internal memory. This may occur, for example, when a teacher or the application program 22 deletes the homework prior to a student’s submission.**[0359]** As previously discussed, one type of document that can be transmitted between a handheld unit and an application program 22 is a “note.” In an embodiment of the

invention, a note is simple text message of up to 255 characters that an application program 22 can create for a handheld unit 16 user to download and view. The intended usage of notes in handheld unit 16 in one embodiment is for instructions for homework activities and sending messages to parents and guardians. A note can be “locked” when it is created by the application program 22. A locked note cannot be deleted by the handheld unit 16 user. It can only be deleted by the teacher using the application program 22. Locked notes are typically used when a teacher wants to make sure a note was read by a parent and not deleted by the student.

[0360] A note may be defined by a markup language document, such as an XML document, that the application program 22 provides to a handheld unit 16 when a GetFile request is submitted by the handheld unit 16. A note document has at least one <note> element. The <note> element has an “id” attribute, and may optionally have a “locked” attribute. The “id” attribute is used to identify the note. Each note the application program 22 creates should be assigned a unique identifier. This identifier is used to track the note from creation to deletion. A GUID is typically used although any id that is unique to the handheld unit 16 user is sufficient. The “locked” attribute can have a value of true or false, and indicates whether or not the note is locked. Notes that are “locked” cannot be deleted by the handheld unit 16 user. The default value for the locked attribute is false. The <note> element includes the optional elements <title>, <sender>, and <date>. The <title> element contains text that specifies the title of the note. The <sender> element contains text that specifies the name of the person that assigned the activity. The <date> element contains text that specifies the date the note was created. The date format is CCYYMMDD. The <note> element also includes the required element <body>, which specifies the content of the note. The text content of the <body> element supports use of the
 element to control line breaks.

[0361] Examples of how the <homework> element is used are as follows:**[0362]** Example #1:

```
<note id="3875a28cd6185eda" locked="false">
  <title>This week's reading</title>
  <sender>Mrs. Smith</sender>
  <date>20050612</date>
  <body>Math: pages 34-49, 61-68<br/>History:
    pages 234-250<br/>English:
    pages 110-121, 143-148</body>
</note>
```

[0363] Example #2:

```
<note id="478ab2819abe46ae" locked="true">
  <title>Note to parents!</title>
  <sender>Mrs. Smith</sender>
  <date>20050616</date>
  <body>Billy has been very disruptive this past week. Please
    call me to schedule a conference to discuss Billy's bad
    behavior. Thank you.</body>
</note>
```

[0364] Attributes**[0365]** id

[0366] A required attribute used to identify the note. Each note an application program **22** creates should be assigned a unique identifier. This id is used to track the note from creation to deletion. A GUID is typically used although any id that is unique to the user of the handheld unit **16** is sufficient.

[0367] locked=[true|false]

[0368] An optional attribute indicating the lock state of the note. Notes that are “locked” cannot be deleted by the user of the handheld unit **16**. Default value is false.

[0369] Content**[0370]** <application>

[0371] A required element that identifies the application program **22**.

[0372] <title>

[0373] An optional text element specifying the title of the note.

[0374] <sender>

[0375] An optional text element specifying the name of the person that sent the note.

[0376] <date>

[0377] An optional element specifying the date the note was created. Date format is CCYYMMDD.

[0378] <body>

[0379] A required text element specifying the content of the note. The text content of this element supports use of the
 element to control line breaks.

[0380] If the application program **22** supports notes, it should provide features to create notes. If the application program **22** supports “locked” notes, then the application program **22** should include features to delete the locked notes.

[0381] When a handheld unit **16** submits a GetFileList request, the application program **22**’s response should include any notes it wants downloaded to the requesting handheld unit **16**. Note files will then be transmitted by the application program **22** to the handheld unit **16**.

[0382] When a note is successfully sent to the handheld unit **16**, the handheld unit **16** will call the ConfirmGetFile API via the responder port **23** with the id of the note the application program **22** assigned. Upon receiving this confirmation, the application program **22** will delete the transmitted note unless it is “locked.”

[0383] Communication between the Service Program **26** and the Handheld Units **16**. According to an embodiment of the invention, the service program **26** (**FIG. 1**) communicates with the handheld units **16** using a compact transport protocol and compact markup language documents. As with the communication between the service program **26** and the application programs **22**, the communication between the service program **26** and the handheld units **16** takes the form of a requests and responses. The format of these requests and responses, as well as the structure of the compact transport protocol, will now be described. The compact transport protocol will sometimes be referred to herein as SDTP.

[0384] Requests. To request a service, a handheld unit **16** constructs a service payload using the compact transport protocol syntax. Next, datagram fragments are created as needed with the “Service” field set to 0x40. These datagrams are then transmitted to a personal area network (PAN) coordinator (such as the base station **11**) using MAC data frames.

[0385] Service request payloads are formed using syntax of the form:

```
SDTP/version + SPACE + service_resource_path + CRLF
[ request-header + CRLF ]
CRLF
[ request-body ]
```

[0386] Message headers for request include:

Header	Description	Example
ri	Identification of the request that is returned in the response. Optional.	ri: 100
ua	User agent. Describes the device and version. Optional.	ua: wasabi/1.0
cl	Content length.	cl: 23

[0387] Example Request:

```
SDTP/1.0 /aown
ri:100
ua:wasabi/1.0
```

[0388] Responses. When the service program **26** responds to a request from a handheld unit **16**, the responses are collected using the MAC data request command. Datagram fragments are then broadcast from the PAN coordinator. The handheld unit **16** receives the fragments and assembles them to extract the service response payload.

[0389] Service responses are returned in the syntax of the form:

```
SDTP/version + SPACE + status_code + SPACE +
status_message + CRLF
[ response-header + CRLF ]
CRLF
[ response-body ]
```

[0390] Message headers for response include:

Header	Description	Example
ru	Identifies the URL of the request the response is intended for.	ru: /asown
ri	Identification of the request that response is intended for. Required when request initially specified an id.	ri: 100

-continued

Header	Description	Example
sv	Server. Describes the server and version fulfilling the request. Optional.	sv: AccelTest/2.4
cl	Content length.	cl: 23

[0391] Example Response:

```
SDTP/1.0 200 OK
ru:/auth
ri:100
```

[0392] User Authentication. If the status code returned for any response is a 401 (Unauthorized) code, the user must authenticate itself for the requested service. A response with this status code has the following forms:

[0393] Example 1: User Authentication

```
SDTP/1.0 401 Unauthorized
ru:/act
ri:101
{auth\app\i 5899128ca5637be36\n AccelTest}
{p\ Enter your student ID:}}
```

[0394] Example 2: Administrator PIN Authentication

```
SDTP/1.0 401 Unauthorized
ru:/act
ri:101
{auth\app\i 5899128ca5637be36\n AccelTest}
{p\ Enter the administrator PIN:}}
```

[0395]

Parameter	Validation	Value
auth	Required element	
a	Optional attribute	If present, user is required to supply the administrator PIN.
nk	Optional attribute	If present, the device should not store the user credentials in the key chain.
app	Required element	
app\i	Required attribute	Application GUID
app\n	Optional attribute	Application Name
p	Required element	Prompt to display to user

[0396] On receipt of this response, a device with a keychain key for the application can automatically submit the authorization request. If a keychain entry is not found, the device will use the prompt to collect the authorization key.

[0397] Request

[0398] To authenticate a user, the following service request is used:

```
SDTP/1.0 /auth
{kc\k application_key(app\i application_id}}
```

[0399]

Parameter	Validation	Value
kc	Key chain element; required	Key supplied by user or device.
kc\a	Optional attribute	If present, key is for administrator authentication.
kc\k	Key chain key attribute; required	User or device supplied credentials for application authentication.
app	Application element; required	
app\i	Application id attribute; required	GUID of the application service handler the key is intended for.

[0400] Example 1: User Authentication

```
SDTP/1.0 /auth
ri:101
cl:31
{kc\k 101{app\i 3d8920a182ef4829}}
```

[0401] Example 2: Administrator PIN Authentication

```
SDTP/1.0 /auth
ri:101
cl:31
{kc\k 101{app\i 3d8920a182ef4829}}
```

[0402] Response

```
SDTP/1.0 200 OK
ru:/auth
ri:101
```

[0403] Result Codes:

Code	Description
200	OK
401	Unauthorized
500	Internal server failure
503	Service unavailable

[0404] Activity. Returns the current live activity.

[0405] Request

```
SDTP/1.0 /act
ri:100
cl:61
{act\i activityid {app\i application_id}
{r\i actionid\q questionnumber\userresponse}}
```

[0406]

Parameter	Validation	Value
act	Activity element	
act\i	Optional	ID of the activity the response is intended for.
act\sub	Activity completed; optional	\sub0 = not submitted, default. \sub1 = submitted as complete
app	Application element	
app\i	Optional: Required if activity data is included	GUID of the application server the activity response is intended for.
r	User response element; optional	
r\i	Action id attribute; optional	ID of the action within the activity.
r\q	Question number or id attribute; optional	ID of question for which the response is intended.
r\	Response text; optional	The user's response to the question
nv	Navigation element; optional	Specifies the users navigation choice.
nv\q	Question number attribute	Specifies direct navigation to a specific question number.
nv\b	Back attribute	Specifies backward navigation to the previous action. Activity specific.
nv\f	Forward attribute	Specifies forward navigation to the next action. Activity specific.

[0407] Example 1: Activity Request with response data.

```
SDTP/1.0 /act
ri:101
cl:54
{act\i 43920afe128ca82a1{app\i 5899128ca5637be36}{r\i4\q1\ 47.3}}
```

[0408] Example 2: Activity Request with response data and navigation to question 5.

```
SDTP/1.0 /act
ri:101
cl:54
{act\i 43920afe128ca82a1{app\i 5899128ca5637be36}
{r\i4\q1\ 47.3}{nv\q5}}
```

[0409] Example 3: Activity Request with credentials supplied

```
SDTP/1.0 /act
ri:101
cl:54
{kc\k a387b1{app\i 3d8920a182ef4829}}
```

[0410] Response

[0411] Activity data is contained in device markup.

[0412] Example 1: Standard response with activity data.

```
SDTP/1.0 200 OK
ru:/act
ri:101
cl:82
activity markup data
```

[0413] Example 2: Response with authentication request.

```
SDTP/1.0 401 Unauthorized
ru:/act
ri:101
cl:82
{auth{app\i 5899128ca5637be36\n AccelTest}
{p\ Enter your student ID:}}
```

[0414] Result Codes:

Code	Description
200	OK
401	Unauthorized
403	Forbidden
500	Internal server failure
503	Service unavailable

[0415] File List. Get a list of the files for the device owner from the current application.

[0416] Request

[0417] Example 1: Get Files Request

```
SDTP/1.0 /files
ri:101
```

[0418] Example 2: Get Files Request with credentials supplied

```
SDTP/1.0 /files
ri:101
cl:54
{kc\k a387b1{app\i 3d8920a182ef4829}}
```

[0419] Response

[0420] Application and File elements are contained in the device markup.

Parameter	Validation	Value
fl	filelist element	
app	Application element	
app\i	Application id attribute; required	GUID of the application server the filelist was served from.
hw	Homework element	
hw\i	Homework id attribute; required	ID of the homework file specified by the application.
t	Homework title; optional	Text specifying the homework title.
no	Note element	
no\i	Note id attribute; required	ID of the note file specified by the application.

[0421] Example 1: Standard Response

```
SDTP/1.0 200 OK
ri:101
{fl{app\i 3d8920a182ef4829}{hw\i ab7291ef3910a12{t\
Chapter 1 Quiz}}{hw\i ab7291ef3910a13{t\
Chapter 1.1 Exercise}}{hw\i ab7291ef3910a13{t\
Chapter 1.1 Exercise}}}{no\i ab7291ef3910a14}}
```

[0422] Example 2: Response with authentication request.

```
SDTP/1.0 401 Unauthorized
ru:/files
ri:101
cl:82
{auth{app\i 5899128ca5637be36\n AccelTest}
{p\ Enter your student ID:}}
```

[0423] Result Codes:

Code	Description
200	OK
401	Unauthorized
500	Internal server failure
503	Service unavailable

[0424] Get File. Return the requested file from the specified application server.

[0425] Request

SDTP/1.0 /getf <u>{f\i file_id}{app\i application server guid}</u>		
Parameter	Validation	Value
f	File element; required	

-continued

SDTP/1.0 /getf <u>{f\i file_id}{app\i application server guid}</u>		
Parameter	Validation	Value
f\i	File id attribute; required	GUID identifying the file to get.
app	Application element; required	
app\i	Application id attribute; required	GUID of the application server the file was served from.

[0426] Example:

```
SDTP/1.0 /getf
ri:101
cl:58
{f\i ab7291ef3910a12{app\i 3d8920a182ef4829}}
```

[0427] Response

[0428] Raw file data is sent in response. For homework and notes, the format corresponds to the compact markup language.

[0429] Example:

```
SDTP/1.1 200 OK
ri:101
cl:293
{no\i 3829a3be2763aa17{dt\ 20050612}{t\ Score Results}
{fr\ Mrs. Smith}{bd\ Congratulations. You scored
85% on the Chapter 1 quiz. Be sure to read the next
chapter, sections 1-4 for tomorrow's class.}}
```

[0430] Result Codes:

Code	Description
200	OK
401	Unauthorized
500	Internal server failure
503	Service unavailable
506	File not found

[0431] Confirm File. Confirm that a file was received successfully.

[0432] Request

SDTP/1.0 /conf <u>{f\i file_id}{app\i application server guid}</u>		
Parameter	Validation	Value
f	File element; required	
f\i	File id attribute; required	GUID identifying the file to get.

-continued

SDTP/1.0 /conf {f\i file_id}{app\i application_server_guid}		
Parameter	Validation	Value
app	Application element; required	
app\i	Application id attribute; required	GUID of the application server the file was served from.

[0433] Example:

SDTP/1.0 /conf ri:101 cl:68 {f\i ab7291ef3910a12{app\i 3d8920a182ef4829}}
--

[0434] Response

SDTP/1.1 200 OK ri:101

[0435] Result Codes:

Code	Description
200	OK
401	Unauthorized
500	Internal server failure
503	Service unavailable
516	File not found

[0436] Put File. Upload a homework file to an application program 22.

[0437] Request

[0438] Homework response data file.

SDTP/1.0 /putf ri:101 {f\i fileid{app\i appid}homework markup}		
Parameter	Validation	Value
f	File element; required	
f\i	Homework id attribute; required	The id identifying the file.
app	Application element; required	
app\i	Application id attribute; required	GUID of the application server the activity response is intended for.
hw	Homework element; required	
hw\sub	Homework submitted attribute, optional	\sub0 = not submitted, default. \sub1 = submitted as complete

-continued

r	User response element; optional	
r\i	Action id attribute; optional	ID of the action within the activity.
r\q	Question number or id attribute; optional	ID of question for which the response is intended.
r\	Response text; optional	The user's response to the question

[0439] Example homework upload:

SDTP/1.0 /putf ri:101 cl:124 {f\i ab7291ef3910a12{app 3d8920a182ef4829}{hw\sub1 {r\i1\q1\ b}{r\i2\q2\ b}{r\i3\q3\ a}{r\i4\q4\ 42.3}{r\i5\q5\ t}}}
--

[0440] Response

SDTP/1.1 200 OK. ri:101

[0441] Result Codes:

Code	Description
200	OK
401	Unauthorized
500	Internal server failure
503	Service unavailable
516	File not found
517	Server is out of storage space

[0442] The wireless audience response system described herein may be used in a variety of scenarios. Similarly, the application programs 22 may be implemented as any of a number of different types of programs. Now, several usage scenarios or “use cases” for the wireless audience response system will be described, with appropriate references to FIG. 1.

[0443] First Run Setup. This use case details the first run setup experience. The application program 22A in this scenario is a “first run” application, which is the application that is invoked after a software installation on the computer 12 is complete. This “first run” application configures the access point 14, guides the teacher through a setup procedure of the handheld unit 16, and performs other “one time, first run” tasks.

[0444] Use case begins.

[0445] 1. Application program 22A launches and requests teacher hook up access point 14.

[0446] 2. Teacher locates access point 14 and hooks it up to the computer 12 via the USB cable.

[0447] 3. Teacher clicks Next button in the installer utility.

[0448] 4. Application program 22A verifies access point 14 is working.

[0449] 5. Application program 22A requests teacher specify name of access point 14.

[0450] 6. Teacher specifies name as Mrs. Jones' Class.

[0451] 7. Application program 22A requests that the teacher power on a handheld unit 16.

[0452] 8. Teacher powers on a handheld unit 16.

[0453] 9. Teacher is shown on screen of application program 22A UI how to connect to wireless network 24 if there is more than one network showing up.

[0454] 10. Application program 22A requests teacher press ENTER button on handheld unit 16.

[0455] 11. Teacher presses ENTER button.

[0456] 12. Application program 22A verifies the handheld unit 16 and access point 14 are working.

[0457] 13. Application program 22A asks if teacher wants to secure the access point 14.

[0458] 14. Teacher indicates she does.

[0459] 15. Application program 22A requests security settings for the access point 14.

[0460] 16. Teacher enters the information.

[0461] 17. Application program 22A indicates the teacher needs to update the network settings on each handheld unit 16 that will connect to this access point 14.

[0462] 18. Teacher updates the settings on each handheld unit 16 as needed.

[0463] Use case ends.

[0464] The application program 22A might include a step that asks whether the teacher wants to assign ownership of devices to students. If the teacher will not be assigning ownership, then setup is complete. If ownership is desired, handheld unit 16 might do it or the application program 22A could do it. Alternatively, student rosters may be stored on the remote computer 28, which the teacher could access via a website and download.

[0465] Network security setup. This use case details the steps for establishing advanced network security settings.

[0466] Use case begins.

[0467] 1. Teacher installs service program 26 and connects access point 14.

[0468] 2. Teacher launches utility to configure service program 26.

[0469] 3. Teacher runs service program 26 using utility.

[0470] 4. Teacher enables security settings, including setting an encryption key of "0ab31def5" to the access point 14.

[0471] 5. Teacher names her network (wireless network 24) Mrs. Smith's Class.

[0472] 6. Teacher quits the utility.

[0473] 7. Teacher powers on handheld unit 16.

[0474] 8. Teacher selects "Security" in "Settings" menu of handheld unit 16.

[0475] 9. Teacher enters encryption key "0ab31def5".

[0476] 10. Teacher selects "Search For Networks" in "Settings" menu.

[0477] 11. Handheld unit 16 performs an active scan for 802.15.4 networks in range.

[0478] 12. Handheld unit 16 locates two networks, Mrs. Smith's Class and Science Lab.

[0479] 13. Handheld unit 16 displays list of network names.

[0480] 14. Teacher selects Mrs. Smith's Class.

[0481] 15. Handheld unit 16 connects to wireless network 24.

[0482] 16. Teacher has confirmed device network settings work correctly.

[0483] Use case ends.

If encryption keys are used, this use case may include the steps to specify these settings as well. Both the access point 14 and handheld unit 16 would include setup steps to input the keys and settings for encryption.

[0484] Power on, shared device. This use case details the steps of powering on a shared handheld unit 16.

[0485] Use case begins.

[0486] 1. Student powers on handheld unit 16.

[0487] 2. Handheld unit 16 checks for existence of ownership of the device.

[0488] 3. Handheld unit 16 determines device is NOT owned.

[0489] 4. Handheld unit 16 scans for 802.15.4 networks in range.

[0490] 5. Handheld unit 16 locates two networks, Mrs. Smith's Class and Science Lab.

[0491] 6. Handheld unit 16 displays list of network names.

[0492] 7. Student selects Mrs. Smith's Class.

[0493] 8. Handheld unit 16 connects to wireless network 24.

[0494] 9. Student selects Join Session from main menu of handheld unit 16.

[0495] 10. Handheld unit 16 sends an "Activity" request to the service program 26.

[0496] 11. Service program 26 responds there is no live activity running.

[0497] 12. Handheld unit 16 displays main screen and waits for ENTER to be pressed.

[0498] Use case ends.

[0499] Owner assignment. This use case describes the process to assign an owner to a handheld unit 16.

[0500] Use case begins.

[0501] 1. Homeroom begins.

[0502] 2. Teacher launches application program 22B, which is assumed to be an in-class testing program.

[0503] 3. Teacher adds each student to the database 18, which is used by the application program 22B, specifying name and other needed properties (grade, id, etc).

[0504] 4. Application program 22B asks the service program 26 to route all owner assignment requests to its service handler API.

[0505] 5. Teacher powers on a handheld unit 16.

[0506] 6. Teacher selects "Assign Owner" from "Settings" menu on handheld unit 16.

[0507] 7. Handheld unit 16 requests service program 26 to provide owner information.

[0508] 8. Service program 26 routes the request to the application program 22B.

[0509] 9. The application program 22B creates a response asking the owner of the handheld unit 16 to provide a Student ID.

[0510] 10. Teacher enters student ID using keypad of handheld unit 16 and presses ENTER.

[0511] 11. Handheld unit 16 requests service program 26 to provide owner information for Student ID.

[0512] 12. Service program 26 posts the request to the application program 22B.

[0513] 13. The application program 22B processes the request, and responds with Student owner information.

[0514] 14. Handheld unit 16 accepts owner information and saves it in memory.

[0515] 15. Teacher calls student to whom the handheld unit 16 was assigned.

[0516] 16. Teacher gives handheld unit 16 to student.

[0517] 17. Student selects "Change PIN" from "Settings" menu on handheld unit 16.

[0518] 18. Handheld unit 16 requests new PIN from Student.

[0519] 19. Student enters PIN.

[0520] 20. Handheld unit 16 records new PIN.

[0521] 21. Student powers off handheld unit 16.

[0522] 22. Handheld unit 16 locks itself with student PIN.

[0523] Use case ends.

[0524] In one embodiment of the invention, when an application program 22 sets the owner information, it will include the key for itself to be stored on the virtual key chain of the handheld unit 16. This way, the student does not need to enter this later when working with application program 22. An application key allows automatic identification of the student/handheld unit 16 when activities and notes are manipulated. The teacher may also elect to label the handheld unit 16 with a waterproof pen or printed label.

[0525] Reset owner. This use case describes the process to reset the owner on a handheld unit 16.

[0526] Use case begins.

[0527] 1. Teacher powers on a handheld unit 16.

[0528] 2. Teacher selects "Reset Owner" from the "Settings" menu.

[0529] 3. Handheld unit 16 requests that the Administrator PIN be entered.

[0530] 4. Teacher enters her Administrator PIN.

[0531] 5. Handheld unit 16 requests that the service program 26 validate the Administrator PIN and reset owner information.

[0532] 6. Service program 26 validates the Administrator PIN and confirms reset.

[0533] 7. Service program 26 responds to handheld unit 16 with reset confirmation response.

[0534] 8. Handheld unit 16 resets itself, clearing owner information, all student data, application keys, PIN, notes, and homework.

[0535] Use case ends.

[0536] Power on, owned device. This use case details the steps of powering on a handheld unit 16 that is owned.

[0537] Use case begins.

[0538] 1. Student powers on handheld unit 16.

[0539] 2. Handheld unit 16 checks for existence of ownership.

[0540] 3. Handheld unit 16 determines that it is owned.

[0541] 4. Handheld unit 16 determines whether an owner PIN is present and was used to lock the device.

[0542] 5. Handheld unit 16 requests student enter the PIN that was used to lock it.

[0543] 6. Student enters PIN and presses ENTER.

[0544] 7. Handheld unit 16 determines PIN entered matches owner PIN used to lock device.

[0545] 8. Handheld unit 16 presents main menu.

[0546] Use case ends.

[0547] Power off, owned device. This use case details the steps of powering off a handheld unit 16 that is owned.

[0548] Use case begins.

[0549] 1. Student presses and holds the power button handheld unit 16.

[0550] 2. Handheld unit 16 checks for existence of ownership of the device.

[0551] 3. Handheld unit 16 determines device is owned.

[0552] 4. Handheld unit 16 determines if an owner PIN is present.

[0553] 5. Handheld unit 16 locks the device before power off. PIN will be required on power on.

[0554] Use case ends.

[0555] Forgotten PIN. This use case details the steps of powering on a handheld unit **16** that is owned and the owner has forgotten their PIN.

[0556] Use case begins.

[0557] 1. Student powers on handheld unit **16**.

[0558] 2. Handheld unit **16** checks for existence of ownership of the device.

[0559] 3. Handheld unit **16** determines device is owned.

[0560] 4. Handheld unit **16** determines if an owner PIN was present and used to lock the device.

[0561] 5. Handheld unit **16** requests student enter the PIN that locked the handheld unit **16**.

[0562] 6. Student enters PIN and presses Enter.

[0563] 7. Handheld unit **16** determines PIN entered does NOT match owner PIN used to lock device.

[0564] 8. Handheld unit **16** informs user PIN is invalid.

[0565] 9. Student informs teacher PIN was forgotten.

[0566] 10. Student hands teacher handheld unit **16**.

[0567] 11. Teacher presses “?” key on the handheld unit **16** screen.

[0568] 12. Handheld unit **16** scans for 802.15.4 networks in range.

[0569] 13. Handheld unit **16** locates two networks, Mrs. Smith’s Class and Science Lab.

[0570] 14. Handheld unit **16** displays list of network names.

[0571] 15. Teacher selects Mrs. Smith’s Class.

[0572] 16. Handheld unit **16** connects to network.

[0573] 17. Handheld unit **16** requests Administrator PIN.

[0574] 18. Teacher enters Administrator PIN.

[0575] 19. Handheld unit **16** requests server validate the PIN.

[0576] 20. Service program **26** validates Administrator PIN.

[0577] 21. Handheld unit **16** receives response from server and displays the student’s PIN.

[0578] Use case ends.

[0579] Anonymous live activity, unprompted. This use case details the experience of a handheld unit **16** initiating communication with the wireless network **24** while a live activity is running. This activity is anonymous, speaker paced, unprompted, and requires no user identification.

[0580] Use case begins.

[0581] 1. Mark launches application program **22C** which, in this use case, is sales presentation software.

[0582] 2. Mark opens presentation he made, which includes a Q&A panel portion that is to be conducted using the handheld units **16**.

[0583] 3. Attendees to Mark’s presentation begin to file into the auditorium.

[0584] 4. Mark’s team hands out a handheld unit **16** to each attendee.

[0585] 5. Mark informs attendees to power on their handheld units **16** and how they will use them.

[0586] 6. Attendee powers on handheld unit **16**.

[0587] 7. Handheld unit **16** checks for existence of ownership of the device.

[0588] 8. Handheld unit **16** determines device is NOT owned.

[0589] 9. Attendee selects “Join Session” from main menu.

[0590] 10. Handheld unit **16** scans for 802.15.4 networks.

[0591] 11. Handheld unit **16** locates one network and connects automatically.

[0592] 12. Handheld unit **16** enters live activity currently in progress.

[0593] 13. Mark begins his sales presentation.

[0594] 14. At the slide on the two product offerings the company is considering, Mark asks attendees to pick the product they thing has the best appeal to customers.

[0595] 15. 73% of respondents select product “A” by pressing A on their handheld units **16** and then pressing the ENTER button.

[0596] 16. 25% of respondents select product “B”

[0597] 17. 163 of the 184 attendees responded.

[0598] 18. Mark takes note of the response and moves on in the presentation.

[0599] 19. At the last slide, Mark asks attendees to press their ENTER buttons a second time, then answer the question on the slide: “Did this presentation provide you with an accurate description of our sales strategy?”

[0600] 20. 84% of respondents indicate Yes by pressing True button on their handheld unit **16** and then pressing the ENTER button.

[0601] 21. 15% of respondents indicate No.

[0602] 22. 180 of the 184 attendees responded.

[0603] 23. Mark senses that too many attendees responded with “No,” so he conducts an ad hoc Q&A session to answer any questions on the strategy.

[0604] 24. Mark accesses application program **22C** to end live activity.

[0605] 25. Attendee presses ENTER button.

[0606] 26. Handheld unit **16** returns to main screen and awaits next live activity.

[0607] 27. Attendee presses ENTER button.

[0608] 28. Handheld unit **16** requests server for live activity.

- [0609] 29. Server returns response indicating there is not a live activity.
- [0610] 30. Handheld unit **16** displays main screen and awaits next live activity.
- [0611] 31. Mark thanks attendees and asks they return their handheld unit **16** to the staff at the exits.
- [0612] 32. Attendee hands handheld unit **16** to staff at exit.
- [0613] 33. Staff powers off handheld unit **16** and places it in box.
- [0614] Use case ends.
- [0615] If a handheld unit **16** has an owner, the live activity will not be entered automatically. The user will instead see a "Live Activity" menu choice to select to participate in the activity. If the device is owned and locked with a PIN, the PIN will be requested on Power On.
- [0616] Anonymous live activity, prompted. This activity is anonymous, attendee paced, prompted, and requires no user identification. This use case also presents a notion of message notes that might be used.
- [0617] Use case begins.
- [0618] 1. Mark launches application program **22D** which, in this use case, is survey software.
- [0619] 2. Mark opens survey he made with the software
- [0620] 3. Mark informs attendees to power on and select "Join Session" to participate in the live survey.
- [0621] 4. Attendee powers on handheld unit **16** and selects "Join Session."
- [0622] 5. Handheld unit **16** enters live activity currently in progress.
- [0623] 6. Mark asks that each attendee fill out the survey on their handheld unit **16**.
- [0624] 7. Sarah, an attendee, reads the first question "How would you rate the speaker?, A: Excellent, B: Good, C: Fair, D: Poor"
- [0625] 8. Sarah presses B and then presses ENTER.
- [0626] 9. Handheld unit **16** requests the server to process the response.
- [0627] 10. Application program **22D** processes the response.
- [0628] 11. Handheld unit **16** displays the next question.
- [0629] 12. Sarah answers the next question.
- [0630] 13. Application program **22D** processes the response.
- [0631] 14. Sarah answers the last question and presses ENTER.
- [0632] 15. Application program **22D** processes the response.
- [0633] 16. Application program **22D** returns a message to display to Sarah.
- [0634] 17. Handheld unit **16** receives the response and displays a message that was sent to her. The note reads "Thank you for your attendance to this year's conference. You can locate presentation materials online at www.presos.com/2005. See you next year."
- [0635] 18. Sarah notes the web site and powers off the handheld unit **16**.
- [0636] Use case ends.
- [0637] Student identification. This use case details the process of identifying a student to a host application and storing the login identification information in a key chain key. The key chain device is available for devices that have been assigned an owner. The key chain remembers the login credentials for each application and is used to automatically identify the student the next time he accesses the system **10**.
- [0638] Use case begins.
- [0639] 1. Teacher launches application program **22E** which, in this use case, is a math quiz program.
- [0640] 2. Teacher configures application program **22E** to host a live quiz activity for the class.
- [0641] 3. Teacher selects the option to save individual student responses for later reporting.
- [0642] 4. Teacher directs students to power on their handheld unit **16** and participate in the application program **22E** live activity.
- [0643] 5. Student powers up handheld unit **16**.
- [0644] 6. Student engages handheld unit **16** in live activity.
- [0645] 7. Handheld unit **16** sends request to service program **26** for live activity.
- [0646] 8. Application program **22E** cannot determine student making request.
- [0647] 9. Application program **22E** requests student login to identify themselves.
- [0648] 10. Handheld unit **16** checks key chain for saved credentials for application program **22E**.
- [0649] 11. Handheld unit **16** does not find saved credentials for application program **22E**.
- [0650] 12. Handheld unit **16** displays login screen, requesting student identification for application program **22E**.
- [0651] 13. Student enters application program **22E** login and presses ENTER button.
- [0652] 14. Application program **22E** authenticates login for student.
- [0653] 15. Application program **22E** responds to handheld unit **16** with current classroom activity for the student.
- [0654] 16. Handheld unit **16** receives the live activity from application program **22E**.
- [0655] 17. Handheld unit **16** stores credentials for accessing application program **22E** in the key chain.
- [0656] 18. Handheld unit **16** displays the appropriate screen for the live activity.
- [0657] Use case ends.

[0658] Teacher paced, prompted live activity. This use case details the steps as a student works with the teacher on a live, paced, and prompted activity.

[0659] Use case begins.

[0660] 1. Teacher launches application program 22B.

[0661] 2. Teacher launches Classroom Activity wizard on the application program 22B.

[0662] 3. Teacher configures options for activity. (teacher paced, Quiz, prompted, no calculator, etc).

[0663] 4. Teacher clicks Start button in wizard UI.

[0664] 5. Teacher directs students to power on their handheld unit 16 and select "Join Session" to participate in the live activity.

[0665] 6. Student powers up handheld unit 16 and joins session.

[0666] 7. Handheld unit 16 requests live activity from service program 26.

[0667] 8. Application program 22B determines student making request.

[0668] 9. Application program 22B determines current live activity for student.

[0669] 10. Application program 22B responds with Question #1.

[0670] 11. Student reads question and presses "C" button.

[0671] 12. Student presses ENTER button.

[0672] 13. Handheld unit 16 requests service program 26 to record answer.

[0673] 14. Application program 22B determines student making request.

[0674] 15. Application program 22B determines student activity from request.

[0675] 16. Application program 22B determines current classroom activity for student.

[0676] 17. Application program 22B determines current activity matches student request activity.

[0677] 18. Application program 22B records answer for question #1.*

[0678] 19. Application program 22B responds to handheld unit 16 with response that answer was recorded, wait for your teacher.

[0679] 20. Handheld unit 16 displays wait prompt and waits for next press of ENTER button.

[0680] 21. Teacher waits while all students answer.

[0681] 22. Teacher stops accepting responses for question #1.

[0682] 23. Teacher reviews answers in application program 22B.

[0683] 24. Teacher presents lesson material for question #2.

[0684] 25. Teacher clicks the Next Question button on the application program 22B.

[0685] 26. Application program 22B displays question #2.

[0686] 27. Teacher directs students to press the ENTER button on their handheld unit 16.

[0687] 28. Student presses ENTER button.

[0688] 29. Handheld unit 16 requests activity from server.

[0689] 30. Application program 22B responds with question #2.

[0690] 31. Handheld unit 16 displays true false question screen to student.

[0691] 32. Student reads question and presses "True" button.

[0692] 33. Student presses ENTER button.

[0693] 34. Handheld unit 16 requests server to record answer.

[0694] 35. Application program 22B records answer for question #2.*

[0695] 36. Teacher waits while all students answer.

[0696] 37. Teacher stops accepting responses.

[0697] 38. Application program 22B displays results screen.

[0698] 39. Teacher directs students to press the ENTER button on their handheld units 16.

[0699] 40. Student presses ENTER button.

[0700] 41. Application program 22B tabulates score for each student.

[0701] 42. Application program 22B responds with quiz results.

[0702] 43. Handheld unit 16 displays results screen to student.

[0703] Use case ends.

[0704] The application program 22A may elect to reject the answer to a question if the question is no longer the current question that the teacher is showing on her screen.

[0705] Teacher paced, unprompted live activity. This use case details the steps as a student works with the teacher on a live paced, unprompted activity in the classroom. Application program 22 is host of the live activity in this case.

[0706] Use case begins.

[0707] 1. Teacher launches application program 22B.

[0708] 2. Teacher launches Teacher Paced Session wizard.

[0709] 3. Teacher configures options for activity. (Quiz, unprompted, no calculator, etc).

[0710] 4. Teacher directs students to power on their handheld units 16 and select "Join Session" to participate in the live activity.

- [0711] 5. Student powers up handheld unit 16 and connects to network.
- [0712] 6. Student engages handheld unit 16 in live activity.
- [0713] 7. Handheld unit 16 requests server for live activity.
- [0714] 8. Application program 22B determines student making request.
- [0715] 9. Application program 22B determines current live activity for student.
- [0716] 10. Application program 22B responds with teacher paced, unprompted activity.
- [0717] 11. Handheld unit 16 displays "ready for your response" screen.
- [0718] 12. Teacher presents lesson material for question #1.
- [0719] 13. Teacher directs students to enter A, B, C, or D for their answer to the question on the board.
- [0720] 14. Student presses "C" button.
- [0721] 15. Student presses ENTER button.
- [0722] 16. Handheld unit 16 requests server to record answer.
- [0723] 17. Application program 22B determines student making request.
- [0724] 18. Application program 22B determines student activity from request.
- [0725] 19. Application program 22B determines current classroom activity for student.
- [0726] 20. Application program 22B determines current activity matches student request activity.
- [0727] 21. Application program 22B records answer for question #1.*
- [0728] 22. Application program 22B responds with a prompt that the answer was recorded and to wait for teacher.
- [0729] 23. Handheld unit 16 displays prompt.
- [0730] 24. Teacher waits while all students answer.
- [0731] 25. Teacher stops accepting responses.
- [0732] 26. Teacher reviews answers using chart UI in application program 22B.
- [0733] 27. Teacher clicks the Next Question button on the application program 22B.
- [0734] 28. Application program 22B displays question #2.
- [0735] 29. Teacher presents lesson material for question #2.
- [0736] 30. Teacher directs students to press ENTER button, then enter True or False for their answer to the question on the board.
- [0737] 31. Student presses "True" button.
- [0738] 32. Student presses ENTER button.
- [0739] 33. Handheld unit 16 requests server to record answer.
- [0740] 34. Application program 22B determines student making request.
- [0741] 35. Application program 22B determines student activity from request.
- [0742] 36. Application program 22B determines current classroom activity for student.
- [0743] 37. Application program 22B determines current activity matches student request activity.
- [0744] 38. Application program 22B records answer for question #2.*
- [0745] 39. Application program 22B responds with a prompt that the answer was recorded and to wait for teacher.
- [0746] 40. Handheld unit 16 displays prompt.
- [0747] 41. Teacher waits while all students answer.
- [0748] 42. Teacher stops accepting responses.
- [0749] 43. Teacher reviews answers using chart UI in application program 22B.
- [0750] 44. Teacher clicks the Done button on the application program 22B UI.
- [0751] 45. Teacher directs students to turn off their handheld units 16.
- [0752] Use case ends.
- [0753] Student paced, prompted live activity. This use case details the steps as a student works with the on an individualized live paced, prompted activity in the classroom. Application program 22 is host of the live activity in this case.
- [0754] Use case begins.
- [0755] 1 Teacher launches application program 22B.
- [0756] 2. Teacher launches Student Session wizard.
- [0757] 3. Teacher configures options for activity. (student paced, Quiz, prompted, shuffle, time limit, no calculator, etc).
- [0758] 4. Teacher directs students to power on their handheld units 16 and join the live activity.
- [0759] 5. Student powers up handheld unit 16 and connects to network.
- [0760] 6. Student engages handheld unit 16 in live activity.
- [0761] 7. Responder requests server for live activity.
- [0762] 8. Application program 22B determines student making request.
- [0763] 9. Application program 22B determines current classroom activity for student.
- [0764] 10. Application program 22B responds with Student Paced/Prompted activity.
- [0765] 11. Handheld unit 16 displays multiple choice question #1 to student.

- [0766] 12. Student reads question and presses “C” button.
- [0767] 13. Student presses ENTER button.
- [0768] 14. Handheld unit 16 requests service program 26 to record answer.
- [0769] 15. Application program 22B determines student making request.
- [0770] 16. Application program 22B determines student activity from request.
- [0771] 17. Application program 22B determines current classroom activity for student.
- [0772] 18. Application program 22B determines current activity matches student request activity.
- [0773] 19. Application program 22B records answer for question #1.
- [0774] 20. Application program 22B responds to handheld unit 16.
- [0775] 21. Handheld unit 16 displays true false question #2 to student.
- [0776] 22. Student reads question and presses “True” button.
- [0777] 23. Student presses ENTER button.
- [0778] 24. Handheld unit 16 requests service program 26 to record answer.
- [0779] 25. Application program 22B records answer and responds to handheld unit 16.
- [0780] 26. Handheld unit 16 displays final question to student.
- [0781] 27. Student reads question and presses appropriate answer button.
- [0782] 28. Student presses ENTER button.
- [0783] 29. Handheld unit 16 displays end of assignment screen.
- [0784] 30. Student uses keys on handheld unit 16 to move back and forth to each question for review.
- [0785] 31. Student selects menu to exit activity.
- [0786] 32. Handheld unit 16 asks student if they are ready to submit the assignment as complete.
- [0787] 33. Student presses the “Yes” key and then presses ENTER key.
- [0788] 34. Handheld unit 16 requests server to submit completed activity.
- [0789] 35. Application program 22B marks activity submitted and completed.
- [0790] Use case ends.
- [0791] Student paced, unprompted live activity. This use case details the steps as a student works with the on an individualized live paced, unprompted activity in the classroom. Application program 22B is host of the live activity in this case.
- [0792] Use case begins.
- [0793] 1. Teacher launches application program 22B.
- [0794] 2. Teacher launches Classroom Activity wizard.
- [0795] 3. Teacher configures options for activity. (student paced, unprompted, time limit, no calculator, etc).
- [0796] 4. Teacher hands out hard copy assignments.
- [0797] 5. Teacher clicks Start button in wizard UI.
- [0798] 6. Teacher directs students to power on their Responder and participate in the live activity.
- [0799] 7. Student powers up handheld unit 16 and connects to network.
- [0800] 8. Student engages handheld unit 16 in live activity.
- [0801] 9. Handheld unit 16 requests live activity.
- [0802] 10. Application program 22B determines student making request.
- [0803] 11. Application program 22B determines current classroom activity for student.
- [0804] 12. Application program 22B responds with student paced, unprompted task.
- [0805] 13. Service program 26 responds to handheld unit 16 with classroom task.
- [0806] 14. Handheld unit 16 disables calculator.
- [0807] 15. Handheld unit 16 displays question #1 screen.
- [0808] 16. Student reads question #1 from hard copy and presses “C” button.
- [0809] 17. Student presses ENTER button.
- [0810] 18. Handheld unit 16 requests host to record answer.
- [0811] 19. Application program 22B determines student making request.
- [0812] 20. Application program 22B determines student activity from request.
- [0813] 21. Application program 22B determines current classroom activity for student.
- [0814] 22. Application program 22B determines current activity matches student request activity.
- [0815] 23. Application program 22B records answer for question #1.
- [0816] 24. Application program 22B responds to handheld unit 16.
- [0817] 25. Handheld unit 16 displays question #2 screen.
- [0818] 26. Student reads question #2 and presses “True” button.
- [0819] 27. Student presses ENTER button.
- [0820] 28. Handheld unit 16 requests service program 26 to record answer.

- [0821] 29. Application program 22B records answer for question #2 and responds to handheld unit 16.
- [0822] 30. Handheld unit 16 displays question #N screen.
- [0823] 31. Student reads question and presses appropriate answer button.
- [0824] 32. Student presses ENTER button.
- [0825] 33. Handheld unit 16 displays question #N+1 screen.
- [0826] 34. Student uses keys on handheld unit 16 to move back and forth to each question for review.
- [0827] 35. Student selects menu to "Submit" completed activity.
- [0828] 36. Handheld unit 16 requests service program 16 to submit activity.
- [0829] 37. Application program 22B marks activity submitted and completed.
- [0830] 38. Application program 22B tabulates score for student.*
- [0831] 39. Application program 22B responds with a results to handheld unit 16.
- [0832] 40. Handheld unit 16 displays results screen to student.
- [0833] 41. Teacher watches application program 22B screen as students complete their activity.
- [0834] 42. Time limit for activity has less than 2 minutes left.
- [0835] 43. Students are informed that the time is almost up.
- [0836] 44. Time limit expires and application program 22B ends the activity.
- [0837] Use case ends.
- [0838] Assessment Master activity.
- [0839] Use case begins.
 - [0840] 1. Teacher launches application program 22F on her computer. In this example, application program 22F is a testing program that interacts with an on-line testing service.
 - [0841] 2. Teacher directs students to power on their handheld unit 16 and participate in the live activity.
 - [0842] 3. Student #1 powers up handheld unit 16 and connects to the wireless network 24.
 - [0843] 4. Student #1 engages handheld unit 16 in live activity.
 - [0844] 5. Handheld unit 16 requests host for live activity.
 - [0845] 6. Application program 22F requests student #1 enter Form ID.
 - [0846] 7. Handheld unit 16 displays screen prompting student #1 to enter Form ID.
 - [0847] 8. Teacher visits each student and provides them with their Form ID.
 - [0848] 9. Student #1 enters the Form ID supplied by the teacher.
 - [0849] 10. Handheld unit 16 provides host with Form ID.
 - [0850] 11. Application program 22F validates Form ID with on-line web service.
 - [0851] 12. Application program 22F requests assessment from on-line web service.
 - [0852] 13. On-line web service locates assessment for the Form ID.
 - [0853] 14. Application program 22F downloads the assessment.
 - [0854] 15. Application program 22F responds to handheld unit 16 with the first question on the assessment.
 - [0855] 16. Handheld unit 16 displays the first question to student #1.
 - [0856] 17. Student #2 enters her Form ID.
 - [0857] 18. Handheld unit 16 provides host with Form ID.
 - [0858] 19. Application program 22F validates Form ID with on-line web service.
 - [0859] 20. Application program 22F checks local cache for assessment.
 - [0860] 21. Application program 22F locates assessment for the Form ID.
 - [0861] 22. Application program 22F responds to handheld unit 16 with the first question on the assessment.
 - [0862] 23. Handheld unit 16 displays the first question to student #2.
 - [0863] Use case ends.
 - [0864] An optional scenario might include the teacher identifying the assessment the entire class will participate in and download it before having students engage in the activity. This might avoid the delay that a student will encounter waiting for the assessment to download after entering a Form ID. If a single assessment is identified, application program 22F might ask for the student ID instead of Form ID. Application program 22F would use the student ID to look up a Form ID if necessary. With this approach, students could use their well known student number that does not change, instead of a Form ID that is unique for each assessment they might take.
 - [0865] Falling behind on a live activity. This use case details the scenario when a student falls behind the live activity and the teacher has moved ahead.
 - [0866] Use case begins.
 - [0867] 1. Student returns his attention to the handheld unit 16 after day dreaming for 15 minutes.
 - [0868] 2. Student reads question # 4 and presses "C" button.
 - [0869] 3. Student presses ENTER button.

- [0870] 4. Handheld unit 16 requests server to record answer.
- [0871] 5. Application program 22B determines student making request.
- [0872] 6. Application program 22B determines student activity from request.
- [0873] 7. Application program 22B determines current live activity for student.
- [0874] 8. Application program 22B determines current activity does match student request activity.
- [0875] 9. Application program 22B responds to handheld unit 16 with question # 12.
- [0876] 10. Handheld unit 16 displays question #12 to student.
- [0877] 11. Student reads question #12 and presses "True" button.
- [0878] 12. Student presses ENTER button.
- [0879] 13. Handheld unit 16 requests server to record answer.
- [0880] 14. Student is now in sync with rest of class.

[0881] Use case ends.

[0882] Expired activity. This use case details the scenario in which a student falls behind or does not finish the live activity and the teacher has quit the classroom application or has moved on to another activity. The live activity that the student was working on is considered expired.

[0883] Use case begins.

- [0884] 1. Student returns his attention to the handheld unit 16 after day dreaming for 15 minutes.
- [0885] 2. Student reads current question and presses "C" button.
- [0886] 3. Student presses ENTER button.
- [0887] 4. Handheld unit 16 requests service program 26 to record answer.
- [0888] 5. Application program 22B determines student making request.
- [0889] 6. Application program 22B determines student activity from request.
- [0890] 7. Application program 22B determines current live activity for student.
- [0891] 8. Application program 22B determines current activity does NOT match student request activity.
- [0892] 9. Application program 22B responds to handheld unit 16 that activity is over.
- [0893] 10. Handheld unit 16 displays message to student.
- [0894] 11. Student acknowledges message.
- [0895] 12. Handheld unit 16 exits activity and displays appropriate screen, such as another live activity or the main screen if no live activity is running, or a login to identify oneself to another application that is now running.

[0896] Use case ends.

[0897] Homework activity. This use case details the creation and assignment of homework activities to students with handheld unit 16. Homework may be given to handheld unit 16 that have been assigned to an owner.

[0898] Use case begins.

- [0899] 1. Teacher launches application program 22B.
- [0900] 2. Teacher launches Homework Activity wizard.
- [0901] 3. Teacher configures options for the homework activity. (Quiz, prompted, shuffle, etc).
- [0902] 4. Teacher clicks Finish button in wizard UI.
- [0903] 5. Application program 22B assigns the homework to each student in the grade book.
- [0904] 6. Teacher directs students to power on their handheld units 16 and synchronize their homework and notes.
- [0905] 7. Student powers up handheld unit 16 and connects to wireless network 24.
- [0906] 8. Student selects "Synchronize" from the main menu.
- [0907] 9. Responder requests list of homework from application program 22B.
- [0908] 10. Application program 22B determines student making request.
- [0909] 11. Application program 22B provides list of new and outstanding homework for student.
- [0910] 12. Handheld unit 16 deletes homework that is no longer on list from its memory.
- [0911] 13. Handheld unit 16 uploads homework that has been completed or updated.
- [0912] 14. Application program 22B receives homework activities from handheld unit 16.
- [0913] 15. Application program 22B marks date of submission on homework. One item is two days late.
- [0914] 16. Application program 22B removes completed homework from the list.
- [0915] 17. Application program 22B retains homework items that are not yet due or complete.
- [0916] 18. Handheld unit 16 requests updated list of homework from application program 22B.
- [0917] 19. Application program 22B provides updated list.
- [0918] 20. Handheld unit 16 verifies completed homework was accepted and removed from application program 22B list.
- [0919] 21. Handheld unit 16 deletes completed homework from its memory.
- [0920] 22. Handheld unit 16 requests the new homework activity from application program 22B
- [0921] 23. Application program 22B provides the new homework activity when requested.
- [0922] 24. Handheld unit 16 provides confirmation to application program 22B that homework was received.

- [0923] 25. Application program 22B marks the homework "Received".
- [0924] 26. Handheld unit 16 completes update of homework.
- [0925] 27. Student powers off handheld unit 16, goes home.
- [0926] 28. Student powers on handheld unit 16.
- [0927] 29. Student selects a homework activity.
- [0928] 30. Student answers questions in activity.
- [0929] 31. Student marks the activity completed.
- [0930] Use case ends.
- Collected activities are deleted automatically off the handheld unit 16. A student may delete homework activities that are not "locked" by the teacher. These might be extra-credit activities.
- [0931] New activity. This use case details the creation of a new homework activity from the student. Homework activities can only be done on a handheld unit 16 that has been assigned to an owner.
- [0932] Use case begins.
- [0933] 1. Teacher launches application program 22B.
- [0934] 2. Teacher starts an answer key in application program 22B with no questions due to time constraints.
- [0935] 3. Teacher directs students to power on their handheld unit 16 and update their homework activities.
- [0936] 4. Student powers up handheld unit 16 and connects to wireless network 24.
- [0937] 5. Teacher informs student to create a new homework activity as she has not had time to set things up in application program 22B.
- [0938] 6. A new activity should include a form ID or some way to identify the assignment in application program 22B.
- [0939] Use case ends.
- [0940] Notes from the teacher. This use case details the handling of Notes sent from the teacher to the handheld unit 16.
- [0941] Use case begins.
- [0942] 1. Teacher launches application program 22B.
- [0943] 2. Teacher selects Johnny from the grade book.
- [0944] 3. Teacher right clicks and selects Notes.
- [0945] 4. Teacher adds a Note to Johnny's parents indicating certain behavior problems Johnny is exhibiting.
- [0946] 5. Class period begins.
- [0947] 6. Teacher directs students to power on their handheld units 16 and synchronize their homework activities.
- [0948] 7. Johnny powers up his handheld unit 16 and connects to the wireless network 24.
- [0949] 8. Johnny selects Synchronize from the main menu.
- [0950] 9. The handheld unit 16 requests file list which includes notes.
- [0951] 10. Application program 22B determines student making request.
- [0952] 11. Application program 22B provides current list of notes for student.
- [0953] 12. Handheld unit 16 deletes notes marked "read" which are no longer on the list from memory, asking user to confirm each delete.
- [0954] 13. Handheld unit 16 requests new notes
- [0955] 14. Application program 22B provides new notes.
- [0956] 15. Handheld unit 16 indicates synchronize is complete.
- [0957] 16. Johnny powers off handheld unit 16, goes home.
- [0958] 17. Johnny powers on handheld unit 16.
- [0959] 18. Johnny shows his parents the Note from Mrs. Jones.
- [0960] 19. Johnny's parents read the note, get angry, and send Johnny to his room.
- [0961] 20. Johnny's parents call the teacher and schedule a conference as directed by the note.
- [0962] 21. Mrs. Smith launches application program 22B.
- [0963] 22. Teacher selects Johnny from the grade book.
- [0964] 23. Teacher right clicks and selects Notes.
- [0965] 24. Teacher deletes the note she now knows was indeed read by Johnny's parents.
- [0966] Use case ends.
- Students can delete Notes that are not locked by the teacher. Locked notes are typically those targeted for parents to read.
- [0967] Lost device replacement. This use case illustrates the replacement strategy for dealing with a lost handheld unit 16. The handheld unit 16 is owned and has homework and notes activities on it.
- [0968] Use case begins.
- [0969] 1. Student reports to teacher that the dog ate his handheld unit 16.
- [0970] 2. Teacher unpacks a new handheld unit 16 from the box and powers it on.
- [0971] 3. Teacher reassigns ownership. See Owner assignment use case.
- [0972] 4. Teacher launches application program 22A.
- [0973] 5. Teacher selects "Synchronize" on handheld unit 16.
- [0974] 6. Handheld unit 16 requests host for homework.
- [0975] 7. Application program 22B provides homework to handheld unit 16.

[0976] 8. Application program 22B provides notes to handheld unit 16.

[0977] 9. Teacher quits application program 22B.

[0978] 10. Teacher hands handheld unit 16 to student.

[0979] 11. Teacher informs student that they will have to make up lost work that they had not turned in.

[0980] 12. Student goes home with handheld unit 16.

[0981] 13. Student works on homework activities, redoing the ones they had started before losing the device.

[0982] 14. Student returns to class the next day and submits homework a day late.

[0983] Use case ends.

[0984] Delete notes. This use case details the steps to delete notes.

[0985] Use case begins.

[0986] 1. Student powers up handheld unit 16.

[0987] 2. Student reads note from last week on reading assignment.

[0988] 3. Student presses the menu key 48 to exit note.

[0989] 4. Handheld unit 16 displays message "Keep this Note?" on the LCD screen 32.

[0990] 5. Student selects "No" and presses the Enter key 46.

[0991] 6. Handheld unit 16 deletes old note.

[0992] Use case ends.

[0993] As noted in previous use cases, only homework and notes that are not "locked" can be deleted by the user. Locked Notes can only be deleted by the teacher or through an owner reset action. Locked homework activities are deleted when completed and submitted to the host that created them, by the teacher, and by an owner reset action.

[0994] It can thus be seen that a new and useful audience response system and method have been described. It should be understood that the illustrated embodiments are examples only, and should not be taken as limiting the scope of the invention. The use of the terms "a" and "an" in the context of describing the invention (especially in the context of the following claims) are to be construed to cover both the singular and the plural. Recitation of ranges of values herein are merely intended to serve as a shorthand method of referring individually to each separate value falling within the range, unless otherwise indicated herein, and each separate value is incorporated into the specification as if it were individually recited herein. All methods described herein can be performed in any suitable order unless otherwise indicated herein or otherwise clearly contradicted by context.

What is claimed is:

1. An audience response system comprising:
 - a base station;
 - a plurality of handheld units; and
 - a wireless network,

wherein the system has at least a first mode and a second mode of operation,

wherein, in the first mode of operation,

the base station transmits a question to at least one of the plurality of handheld units over the wireless network,

the handheld unit displays the question to a user, and

in response to the user's input, the handheld user transmits the user's answer to the question to the base station over the wireless network;

wherein, in the second mode of operation,

the base station transmits an answer template corresponding to the question, but not the question itself, to at least one of the plurality of handheld units over the wireless network,

the handheld unit displays the answer template to the user, and

in response to the user's input, the handheld user transmits the user's answer to the question to the base station over the wireless network.

2. The response system of claim 1,

wherein, in the first mode of operation, the base station transmits the question within a markup language document, and

wherein, in the second mode of operation, the base station transmits the answer template within a markup language document.

3. The response system of claim 2,

wherein the markup language document containing the question in the first mode is a compact markup language document generated based on an extensible markup language document, and

wherein the markup language document containing the answer template in the second mode is a compact markup language document generated based on an extensible markup language document.

4. The response system of claim 1,

wherein, in both the first mode and second mode of operation, the handheld unit receives an input of data uniquely identifying the user, and

wherein, in both the first mode and second mode of operation, the handheld unit transmits the identifying data to the base station along with the answer.

5. The response system of claim 4, further comprising a virtual keychain associated with the user,

wherein the identifying data is stored on the virtual keychain, and

wherein, in both first mode and second mode of operation, the handheld unit receives the identity data directly from the virtual keychain.

6. The response system of claim 1,

wherein, in both the first mode and second mode of operation, the base station displays a user interface representing the answers received from the plurality of handheld units.

7. A method for soliciting responses to questions from an audience over a wireless network, the method comprising:

receiving input indicating that a first mode of operation is being selected,

in the first mode of operation, performing steps comprising:

transmitting a question to at least one of the plurality of handheld units over the wireless network,

displaying the question on the handheld unit to a user, receiving an input response on the handheld unit from the user, and

receiving, over the wireless network, a message comprising the user's input response; and

in the second mode of operation, performing steps comprising:

an answer template to at least one of the plurality of handheld units over the wireless network,

providing, in a non-electronic form, the question to a user of the handheld unit,

electronically displaying the answer template on the handheld unit to the user,

receiving, on the handheld unit, the user's input response to the question, and

receiving, over the wireless network, a message comprising the user's input response.

8. The method of claim 7, wherein, in the second mode of operation, the providing step comprises speaking the question aloud to the plurality of people.

9. The method of claim 7, wherein, in the second mode of operation, the providing step comprises giving a handout having the question printed thereon to each of the plurality of people.

10. The method of claim 7, wherein, in the second mode of operation, the providing step comprises displaying the question on a screen visible to each of the plurality of people.

11. The method of claim 7, wherein, in the first mode of operation,

the question transmitted by the base station is transmitted within a markup language document,

the displaying step comprises the handheld unit processing the markup language document and displaying the question on a display of the handheld unit in accordance with instructions contained in the markup language document,

the handheld unit receives the user's input in a form defined by the markup language document, and

in the second mode of operation,

the answer template transmitted by the base station is transmitted within a markup language document,

the displaying step comprises the handheld unit processing the markup language document and displaying the answer template on a display of the handheld unit in accordance with instructions contained in the markup language document, and

the handheld unit receives the user's input in a form defined by the markup language document.

12. The method of claim 11,

wherein, in the first mode,

the question is one of a sequence of questions contained within the markup language document,

the displaying step comprises displaying the next question in the sequence only when the handheld unit receives a message via the wireless network indicating permission to advance to the next question.

13. The method of claim 11,

wherein, in the first mode,

the question is one of a sequence of questions contained within the markup language document,

the displaying step comprises displaying the next question in the sequence when prompted by the user.

14. The method of claim 11,

wherein, in the first mode,

the question is one of a sequence of questions contained within the markup language document, and the markup language document includes data indicating whether the user is permitted to display a previously answered question in the sequence.

15. A method for transmitting questions to and receiving answers from a plurality of users in an audience, each of the plurality of users possessing a handheld unit that is in communication, via a wireless network, with a response system comprising a base station, the method comprising:

the base station performing steps comprising:

transmitting, over the wireless network, a markup language document to a handheld unit of the plurality of handheld units, the markup language document comprising a question;

receiving a response to the question from the handheld unit over the wireless network;

the handheld unit of the plurality of handheld units performing steps comprising:

displaying the question to the user in possession of the handheld unit,

receiving, from the user in possession of the handheld unit, an input response; and

transmitting the input response over the wireless network to the base station.

16. The method of claim 15,

wherein the response system further comprises a database,

wherein the base station performs further steps comprising:

obtaining the question and a correct answer from the database, and

comparing the response received from the handheld unit with the correct answer.

17. The method of claim 15,

wherein the response system further comprises a remote computer that transmits the question to the base station via a public network.

18. The method of claim 15, wherein
the plurality of users are students,
the base station is located in a classroom,
the question is one of a plurality of questions contained in
the markup language document, and the plurality of
questions represents a homework assignment,
the handheld unit of the plurality of handheld units is
possessed by a student of the plurality of students,
the student performs steps comprising:
inputting the response to the handheld unit outside of
the classroom while the handheld unit is not in
communication with the response system,
repeating the inputting step for each of the questions of
the plurality of questions, thereby creating a plurality
of responses and completing the homework assign-
ment,
entering the classroom,
after which, the handheld unit possessed by the student
performs further steps comprising:
reconnecting to the response system,
transmitting the plurality of responses to the base
station, thereby turning in the completed homework
assignment.

19. The method of claim 15, wherein
the plurality of users are students,

the handheld unit is possessed by a student of the plural-
ity,
the base station is operated by an instructor,
the base station transmits another markup language docu-
ment that contains a note from the instructor to the
parents or guardians of the student, the markup lan-
guage document indicating that the note may not be
deleted by the student, and
the handheld unit blocks an attempt by the student to
delete the note.

20. The method of claim 15, wherein the plurality of users
are students and the question is one of a sequence of
questions contained in the markup language document, the
method further comprising

the base station displaying a user interface to an instructor,
the instructor selecting, via the user interface, an activity
for the students to perform,
in response to the instructor's selection, the base station
generating the markup language document, and
the handheld unit displaying a the plurality of questions in
the sequence as specified by the markup language
document.

21. The method of claim 20, wherein the selected activity
is a quiz, and the instructor controls, via the base station, the
pace at which the handheld unit displays the plurality of
questions of the sequence.

* * * * *