High Level Design

Andrew Atutis Joe Belmont Steve Deyo Ryan Edgar Jon Eggleston
John Ellis Jacor Finlayson Michael Hadley James Licata Brett Peplowski

October 27, 2009

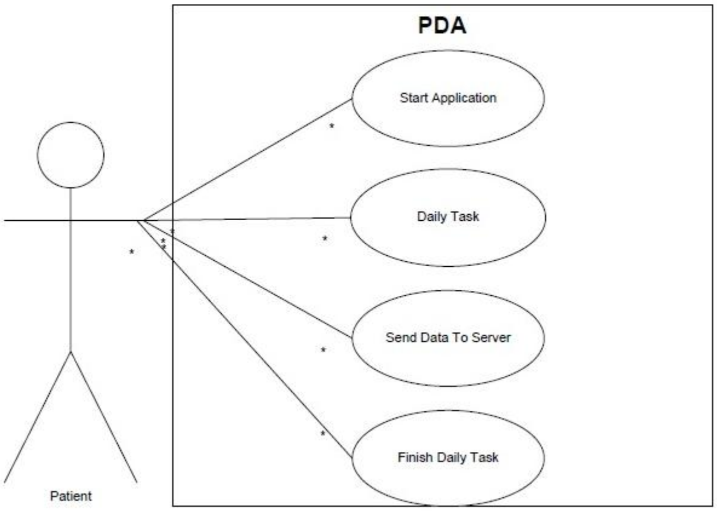
Contents

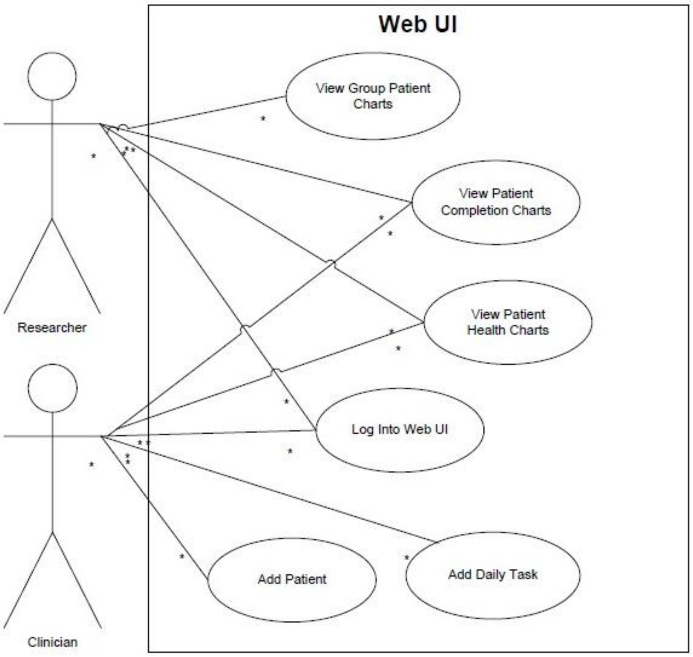
1 Introduction

This document states the high level design for the Mobile Pedometer system. This document should be used by the designers and developers as a basis for the implementation of the system, as it also contains system level use cases, and state diagrams. The goal of this high level design document is to make the reader knowledgeable on the structural design of the system, as well as more knowledgeable about what the system does/is capable of doing. The Activity Monitor system will provide a better way to monitor patient rehabilitation using a Wii Remote and a PDA (Palm Treo 750). The system will be targeted towards nursing homes and rehab hospitals which require monitoring of elderly people, recovering athletes, and people with balance issues. The system will further provide independent mobility while monitoring a patients progress with prescribed Daily Tasks. To perform this, we have developed a Web UI, complete with a database application, which is easy to use and can safely and securely store a patients data.

2 Use Cases

The diagram below shows all of the interactions each user will be able to do with the system once they log in or power on. Patients will interact with the PDA, while Researchers and Clinicians interact with the Web UI. All patients have the ability to start the PDA application, begin their daily task, upload data to the server, and complete their daily task. Researchers and Clinicians share many uses of the system, including Viewing Patient Health Charts, Logging into the Web UI, and Viewing Patient Completion Charts. The Clinician has some accesses to the system that a Researcher does not however. The Clinician may Add Patients to the Web UI as well as assign Daily Tasks. Researchers may view Group Patient Health Charts to compile trends.




Web UI
View Group Patient Charts
View Patient Completion Charts
View Patient Health Charts
Log Into Web UI
Add Patient
Add Daily Task
Researcher
Clinician

3 Scenarios

The following describe common paths that a user may take through the system.

3.1 Patient goes through their Daily Task

The system will provide the following features:

1. The Patient starts the Activity Monitor application on the PDA.
2. The PDA begins to monitor walking activity.
3. The Patient completes the Daily Task and is presented with an alert on the PDA.
4. The PDA begins to download the computed PDA Statistics to the server. The PDA also updates the Daily Task for the next day.
5. The patient exits the Activity Monitor application.

3.2 Clinician adds a new patient

The system will provide the following features:

1. A Clinician logs onto the web interface.
2. The Clinician navigates to the Add Patient page.
3. The Clinician enters the relevant Patient Profile information.
4. The Clinician submits the form and the Patient is added to the database.
5. The Clinician logs off the Web UI.
6. The Clinician configures the PDA so that it will log in the patient with their unique Username and Password.

3.3 A Clinician adds a new task for a patient

The system will provide the following features:

1. A Clinician logs onto the Web UI
2. The Clinician navigates to the List Patients page.
3. The Clinician then clicks on the appropriate Patient and the Patients profile information is displayed.
4. The Clinician clicks on the New Daily Task button.
5. The Clinician enters the appropriate data.
6. The Clinician logs off the Web UI.

3.4 A Clinician reviews Patient Completion Charts/Patient Health Charts

The system will provide the following features:

1. A Clinician logs onto the Web UI
2. The Clinician navigates to the List Patients page.
3. The Clinician clicks on the appropriate Patient.
4. The Clinician clicks on the Patient Health Chart or the Patient Completion Chart
5. The correct chart is displayed with the correct statistical data.
6. The Clinician analyzes the data and discusses with the Patient.
7. The Clinician logs off the Web UI.

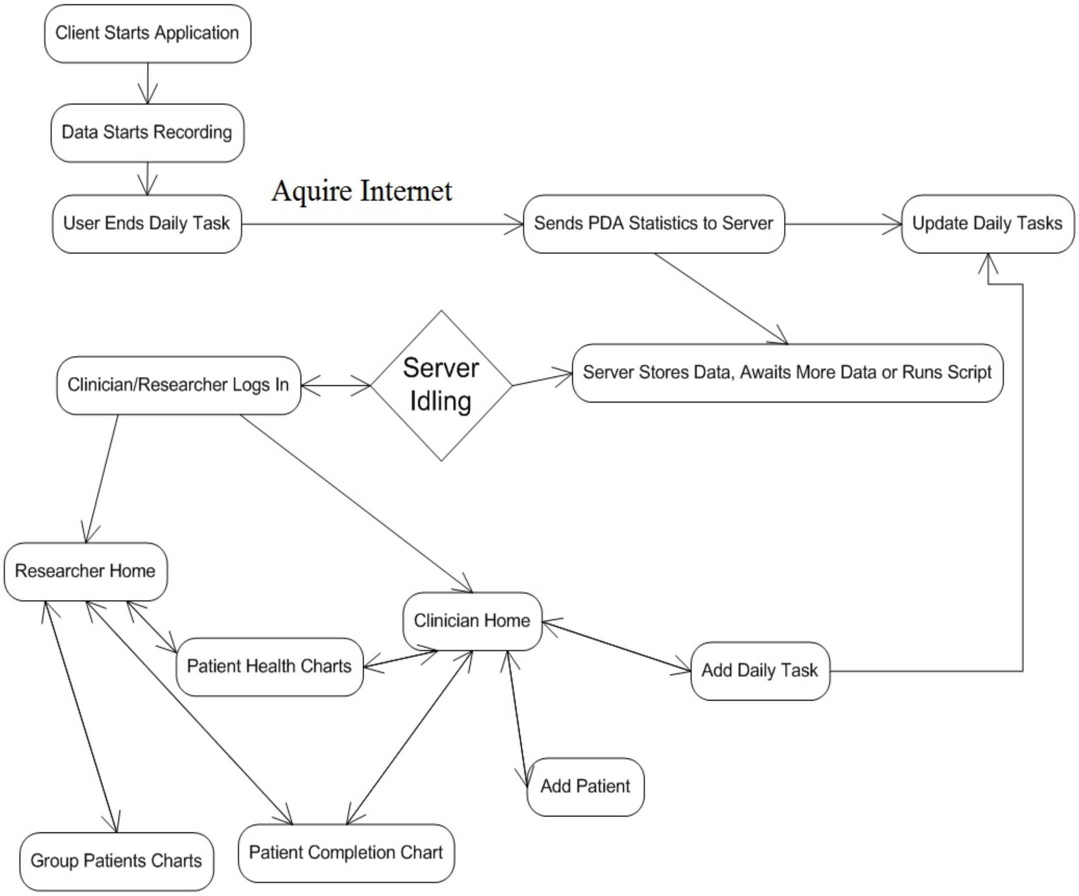
3.5 A Researcher reviews Patient Completion Charts/Patient Health Charts/Group Patient Charts

The system will provide the following features:

1. A Researcher logs onto the Web UI
2. The Researcher navigates to the Patient Completion Chart/Patient Health Chart/Group Patient Chart.
3. The Researcher analyzes the data and reports back to the appropriate Clinicians.
4. The Researcher logs off the Web UI.

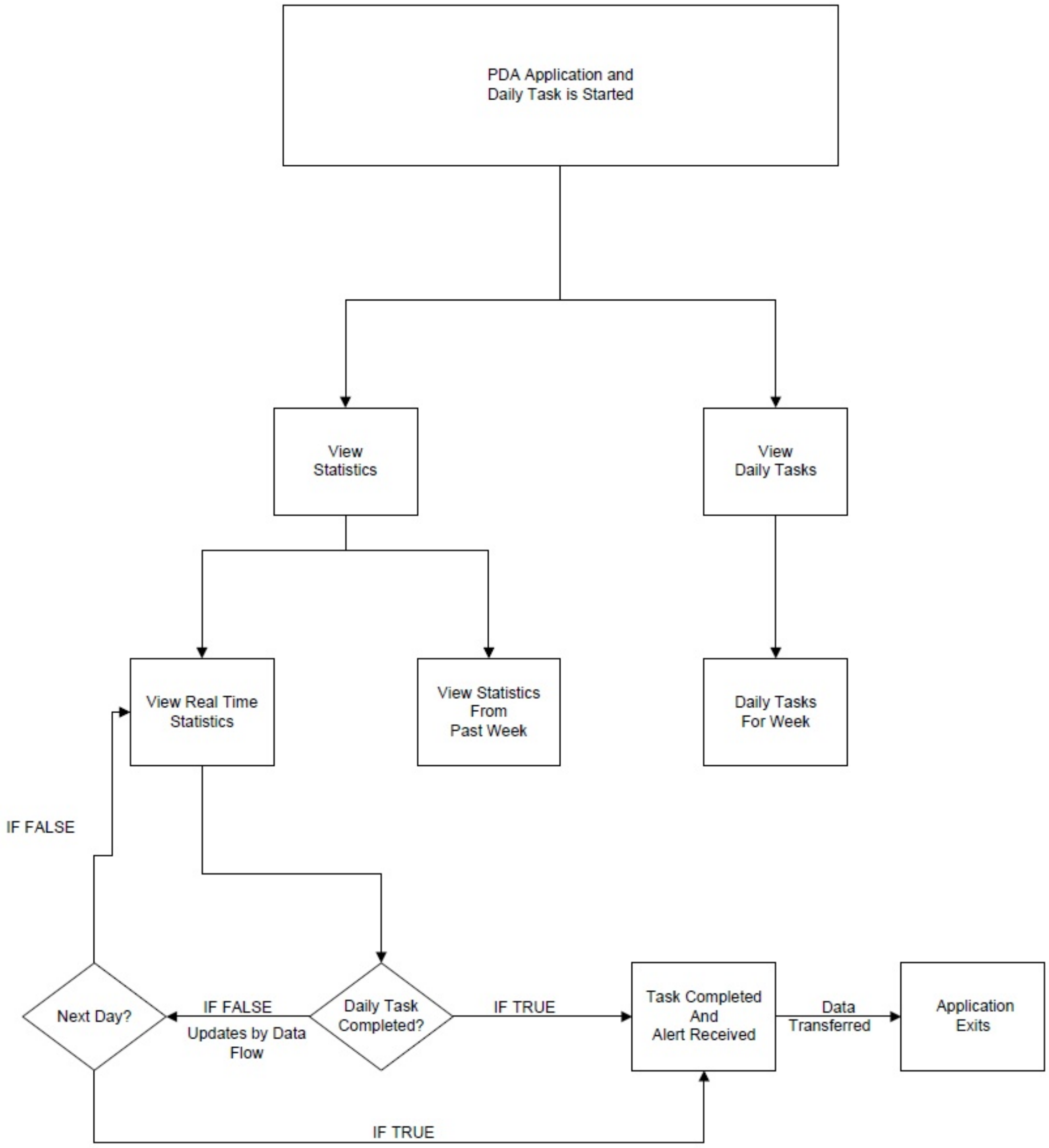
4 System Flow

The following diagram shows how the overall system will function.



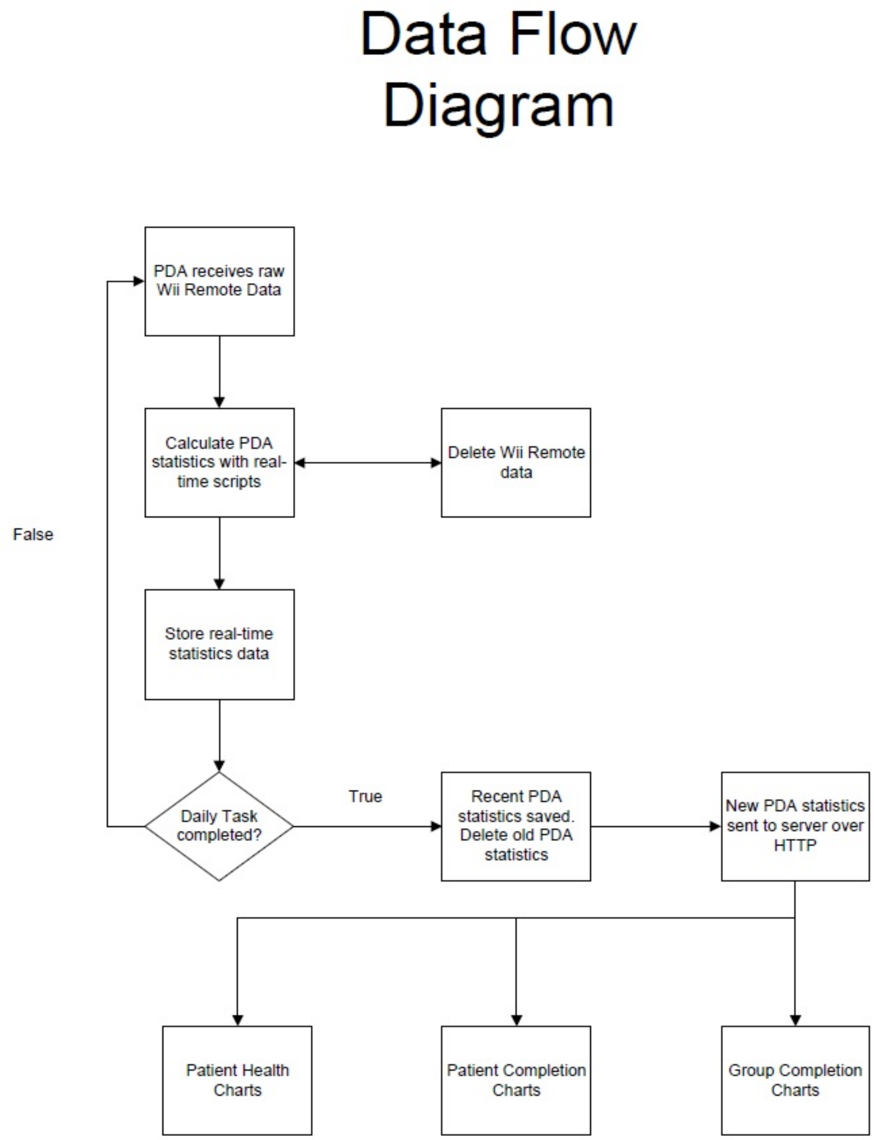
5 User Flow

The following diagram shows the paths for the user interface.



6 Data Flow

The following shows the data flow through the application and server.


Data Flow Diagram
PDA receives raw Wii Remote Data
Calculate PDA statistics with real-time scripts
Delete Wii Remote data
False
Store real-time statistics data
Daily Task completed?
True
Recent PDA statistics saved. Delete old PDA statistics
New PDA statistics sent to server over HTTP
Patient Health Charts
Patient Completion Charts
Group Completion Charts

7 Major System Components

The follow details the major components of the Activity Monitor system. For a illustration see Figure 1: Major System Components Diagram

7.1 PDA

Client GUI

Wii Remote Connection This component represents the software entity that will reside on the PDA. It is responsible for establish a Bluetooth connection with the Wii Remote and retrieving the Raw Wii Remote Data.

Server Connection This component represents the software entity residing on the PDA which will authenticate with the server, and transmit PDA Statistics to the server. This component will also manage updates to the Patient's Daily Tasks.

7.2 Server

PDA Connection This component will reside on the Server and will be responsible for managing the transfer of data to and from the PDA. This component will talk to the Server Connection component on the PDA. These two entities will performing a handshaking protocol (during authentication) as well as transmit PDA Statistics and Daily Task updates.

Web User Interface This component will represent the software component that encompasses the three-tiered Web Application which will reside on the Server. The Web UI will contain a Client User Interface, Application Logic, and a Database.

Application Logic This software entity will fit between the Client User Interface and the Database. It will process Client requests and access the appropriate information from the Database.

Analysis This entity will be contained in the Application Logic. It will deal with processing Client requests pertaining to Clinician and Researcher Charts. The module will retrieve the appropriate information from the Database in order to present the requested chart to the Clinician or Researcher.

Authentication This entity will be contained in the Application Logic. It will deal with processing Researcher/Clinician/Patient Login information. It will verify each user's credentials and report back an appropriate message to the user.

Administration This entity will be contained in the Application Logic and will deal primarily with Adding/Removing/Editing Clinicians, Researchers, and Patients, Modifying Daily Tasks, and shifting Patients between Clinicians.

Database Access Objects (DAOs) This entities will be used to directly modify the Database Object Models. This represents the final stage of the Application Logic and provides an avenue to actually access the Database.

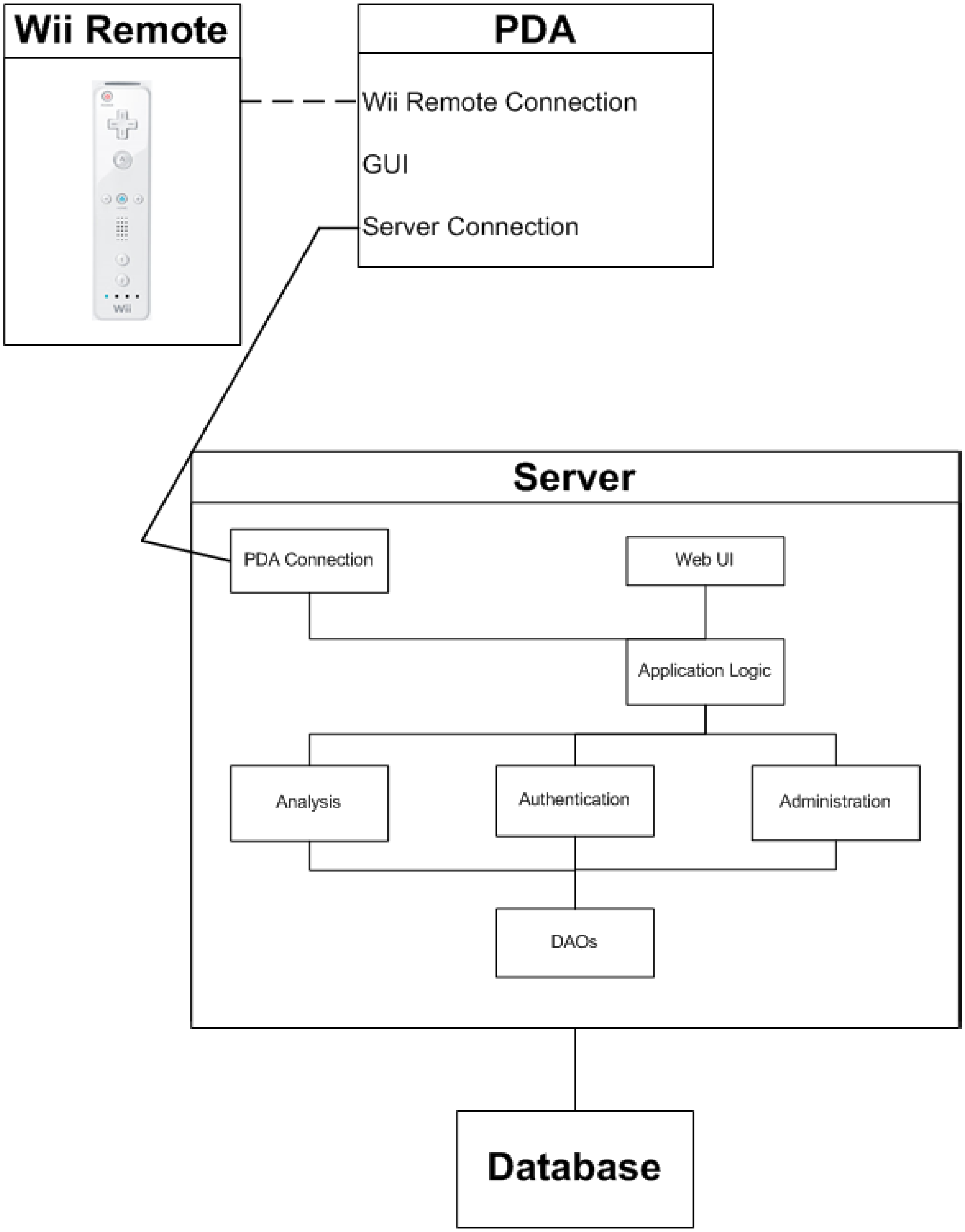
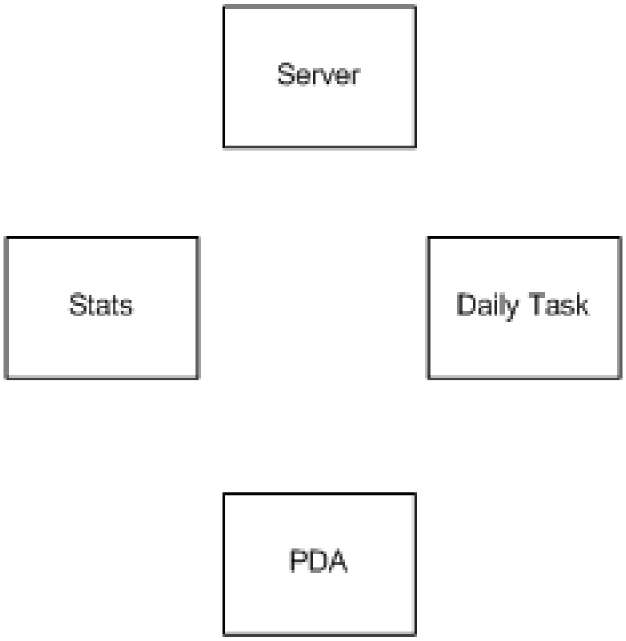


Figure 1: Major System Components Diagram

8 Object Oriented Domain Analysis

8.1 Common Object Architecture



DailyTask

Description This object represents the patient's daily task. The object is known to both the Server and the PDA. This allows both system components to perform operations and access the daily task. The daily task object can represent in theory a wide variety of activities. The simplest being to walk a predetermined number of steps. The Server to PDA interface provides a means of transfer the data associated with a daily task object from the Server to the PDA.

Requirements

- Modify Patient Daily Tasks
- PDA Updates the Patient's Task
- Viewing Tasks for Day

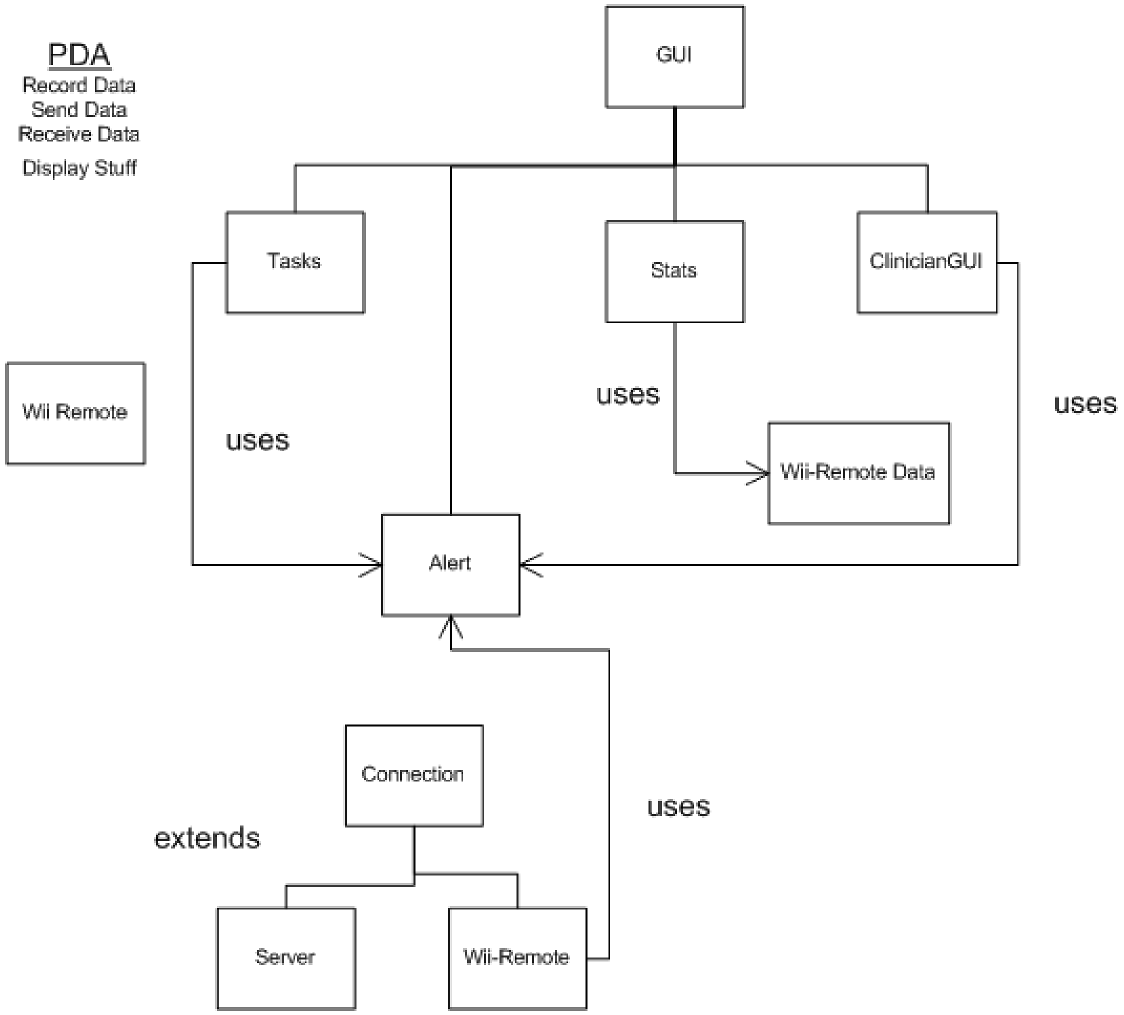
Stats

Description The Stats object is associated with a given sampling period, over which certain statistics are computed by the PDA. The object is known to both the PDA and the Server which allows the statistics to be transferred from the PDA to the Server. This will facilitate organization of the collected data within the database on the Server. It also provides an irreducible unit of sampling time of which the graphs and charts can be constructed.

Requirements

- PDA Displays PDA Statistics in Real Time
- Patient Health Chart
- Group Patient Charts
- Patient Completion Charts

8.2 PDA Object Architecture



8.2.1 Wii Remote Handling

Connection The Connection object provides an interface for the PDA to establish connection with the Wii-Remote and determine the status of the Wii-Remote.

WiiRemote

Description This object encompasses all interaction with the Wii-Remote. This includes connection, and the data obtained from the Wii-Remote.

Requirements

- PDA to Wii Remote Connection
- Connection mode: paired, discoverable

- Detection of Lost Connection by PDA
- Battery Status
- Vibrating on Wii Remote

WiiRemoteData

Description The WiiRemoteData object provides an interface to receive and interpret data from the Wii-Remote to the PDA.

Requirements

- PDA records data from Wii Remote
- PDA Displays PDA Statistics in Real Time

8.2.2 Server Connection Handling

Connection

ServerConnection

Description The ServerConnection object provides an interface for the PDA to connect with the Server to send and receive data. Data that has been collected by the PDA from the WiiRemote will be sent to the Server, while the PDA will update the Patients Daily Task according to the Servers data being sent back to the PDA.

Requirements

- Data Dump
- Emergency Data Dump
- PDA to Server Connection
- PDA Updates the Patient's Task

Stats This is a shared object between the PDA and the Server architectures. For more information see Section 8.1: Common Object Architecture.

DailyTask This is a shared object between the PDA and the Server architectures. For more information see Section 8.1: Common Object Architecture.

8.2.3 Client General User Interface

StatsView

Description The StatsView object interprets Alert Messages for when the Daily Task is completed and displays them to the user, while also displaying real time PDA statistics to the user.

Requirements

- Alert When the Daily Task is Completed
- PDA Displays PDA Statistics in Real Time

DailyTaskView

Description The DailyTaskView object displays Daily Tasks to the user as interpreted from the data collected from the server in easy to read font on the PDAs display.

Requirements

- Viewing Tasks for Day
- Daily Tasks Displayed in a Large, Easy to Read Font

ClinicianGUI

Description The ClinicianGUI object provides an interface for the clinician to enter the information required to connect to the server. This includes the server address, the patient's username, and the patient's password. The information entered on this screen is known only to the clinician. Access to this screen should be obscured from the patient's general activities on the PDA.

Requirements

- PDA to Server Connection

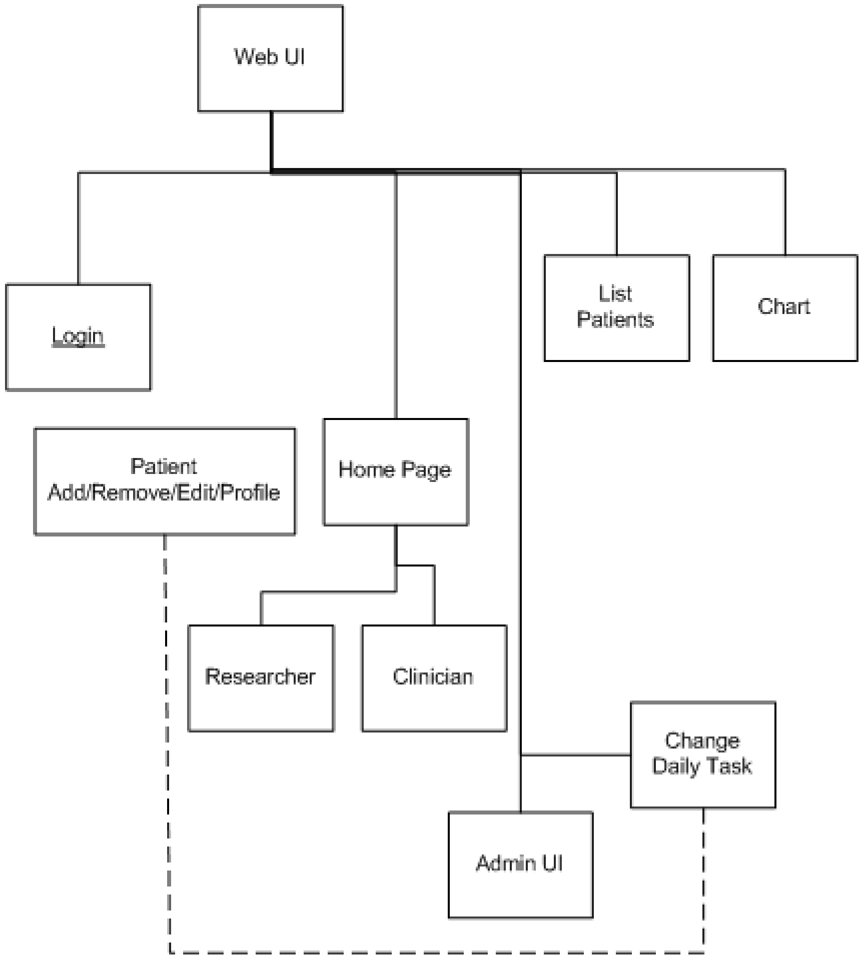
Alert

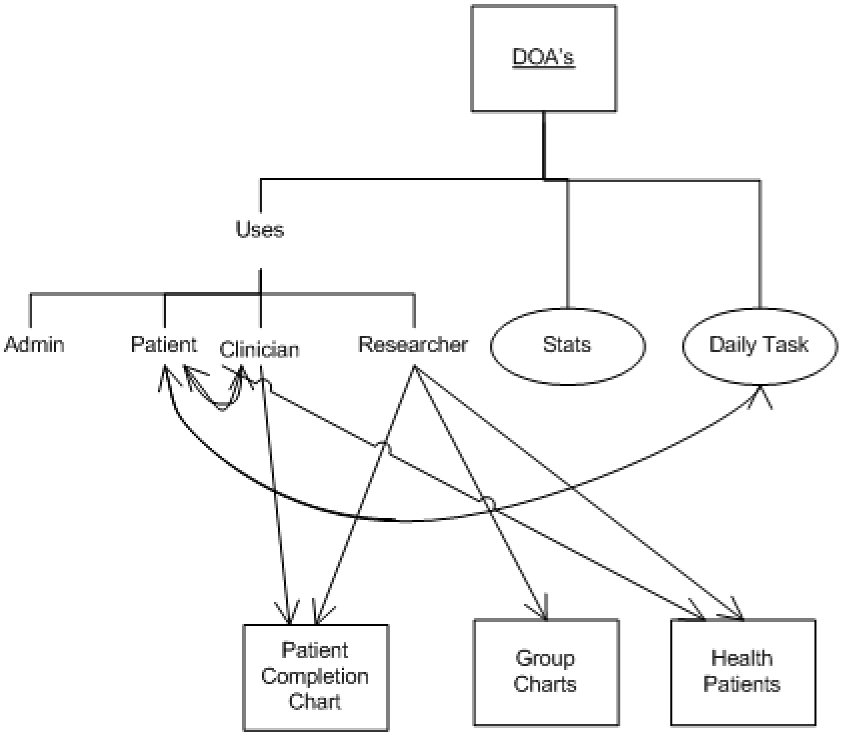
Description This object will deal with all the alerts provided to the Patient on the PDA. This includes alerts for when the Patient completes an exercise, a connection is lost between the PDA and the Wii Remote, and when the PDA goes into Emergency Data Dump Mode.

Requirements

- Emergency Data Dump
- Alerts on PDA Including Text and Audio
- Detection of Lost Connection by PDA
- Battery Status
- Alert When the Daily Task is Completed

8.3 Server Object Architecture





8.3.1 PDA Interaction

Server

Description This is the primary control object on the server side. It will maintain interactions with multiple PDAs and handle any and all administrative and authentication related issues.

PDA

Description This object represents all interaction with the PDA from the Server. It encompasses all of the connection interfaces, as well as PDA Statistics data exchange, and the front-end GUI portion.

Requirements

- Data Dump
- Emergency Data Dump
- PDA to Server Connection
- PDA Updates the Patient's Task

Stats This is a shared object between the PDA and the Server architectures. For more information see Section 8.1: Common Object Architecture.

DailyTask This is a shared object between the PDA and the Server architectures. For more information see Section 8.1: Common Object Architecture.

8.3.2 Web User Interface

This directly fulfills the requirement: Web UI.

Login

Description This object represents all of the logins on the Web UI by both Clinicians and Researchers. It is responsible for querying the Application Logic's Authentication module for verification of the user's credentials.

Requirements

- Researcher Login on Web UI

PatientProfile

Description This object represents a Patient's account in the Web UI. This object has a many to one relationship with Clinicians. It contains all of the relevant Patient information, as well as a Username and Password for the PDA.

Requirements

- Patient Profile
- Unique Usernames for Clinicians and Patients
- List of Available Patients

HomePage

ResearcherHome

ClinicianHome

AdminUI

Description This object will represent the user interface presented to the Administrator in order to Add or Remove Researcher and Clinician user accounts in the Web UI.

Requirements

- Administrator can Add/Remove/Edit Researcher and Clinician Web UI Accounts
- Unique Usernames for Clinicians and Patients
- Administrator Homepage

ChangeDailyTask

Description This object is responsible for changes to a specific Patient's Daily Tasks. It will query the Web UI's Application Logic unit's Administration module to modify the database.

Requirements

- Modify Patient Daily Tasks

ListPatients

Description This object will be accessible to users through the web user interface. It should provide a list of available patients for the currently logged in user as a web page output. This list must include only patients that are under the control of the clinician logged in. Security on this page is of the utmost concern.

The page generated by this object should be simple to understand and easy to navigate. The patient listing that is generated should provide links to each patient's profile.

Requirements

- Patient Profile
- List of Available Patients

Chart

Description This object will be an abstract base for the Researcher and Clinician Charts.

Requirements

- Patient Health Chart
- Group Patient Charts
- Patient Completion Charts

CompletionChart

Description This chart is available to both Clinicians and Researchers and represents a if particular Patient is completing their Daily Tasks.

Requirements

- Patient Completion Charts

HealthChart

Description This chart is available to both Clinicians and Researchers and represents how a particular Patient's Health has improved over the course of their treatment.

Requirements

- Patient Health Chart

GroupChart .

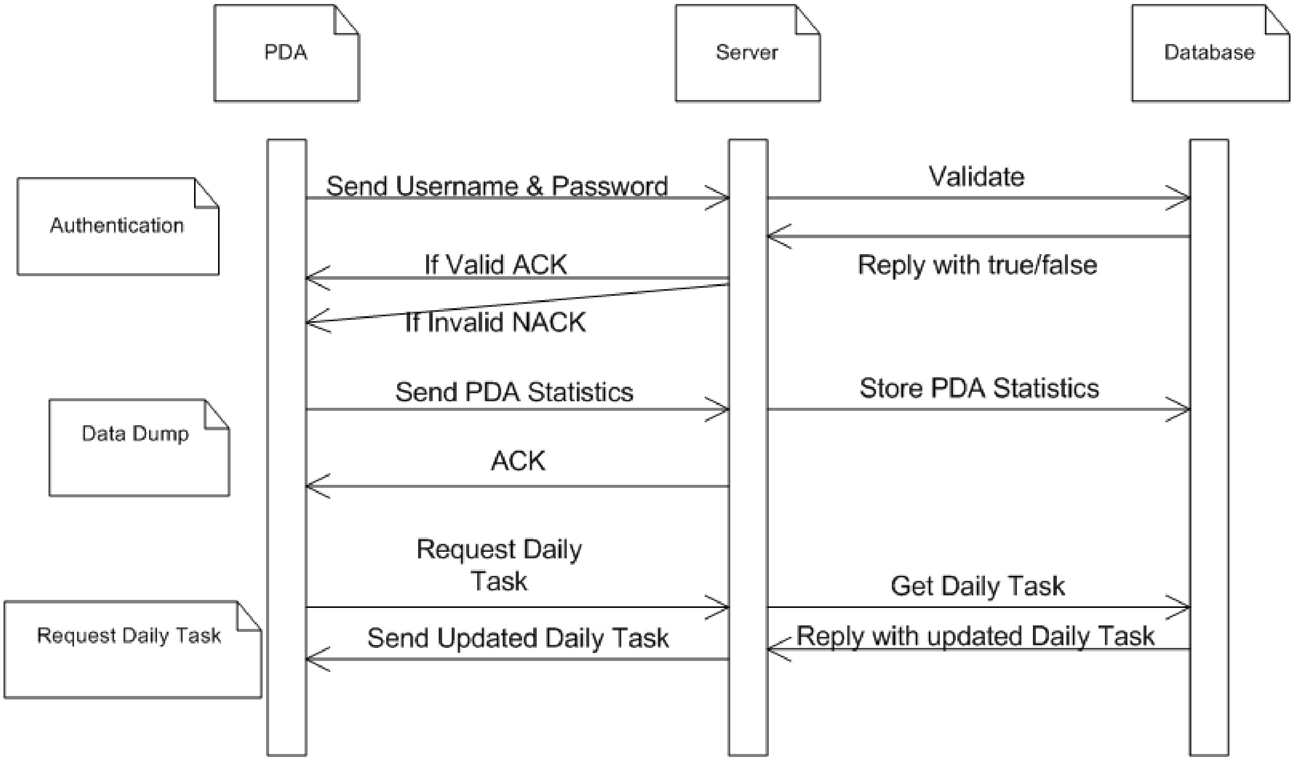
Description This chart is only available to the Researcher and represents trends pertaining to groups of Patient's Daily Tasks.

Requirements

- Group Patient Charts

9 Object Interface Specification

9.1 PDA to Server Interfaces



Authentication

Description The main purpose of the authentication is to verify that the correct user is trying to access the patients information. At the authentication screen, the user will enter their username and password. The username and password that was entered is sent to the server. If the information is correct ACK is sent back by the server. If not, NACK is sent back, and the connection is closed. Once the information is verified, the validation is sent to from the server to the database.

Trigger This task occurs when the PDA is handshaking with the Server during the initialization of a connection. This connection could either be triggered by the need to download the daily task or the need to dump data to the server.

Data Dump

Description The data dump is used to send the information stored on the PDA for the day to the database. At the end of the day, the data is sent over an open connection to the server. When the server receives the information, an ACK is sent back. Once this is done, the information is sent from the server and stored into the database.

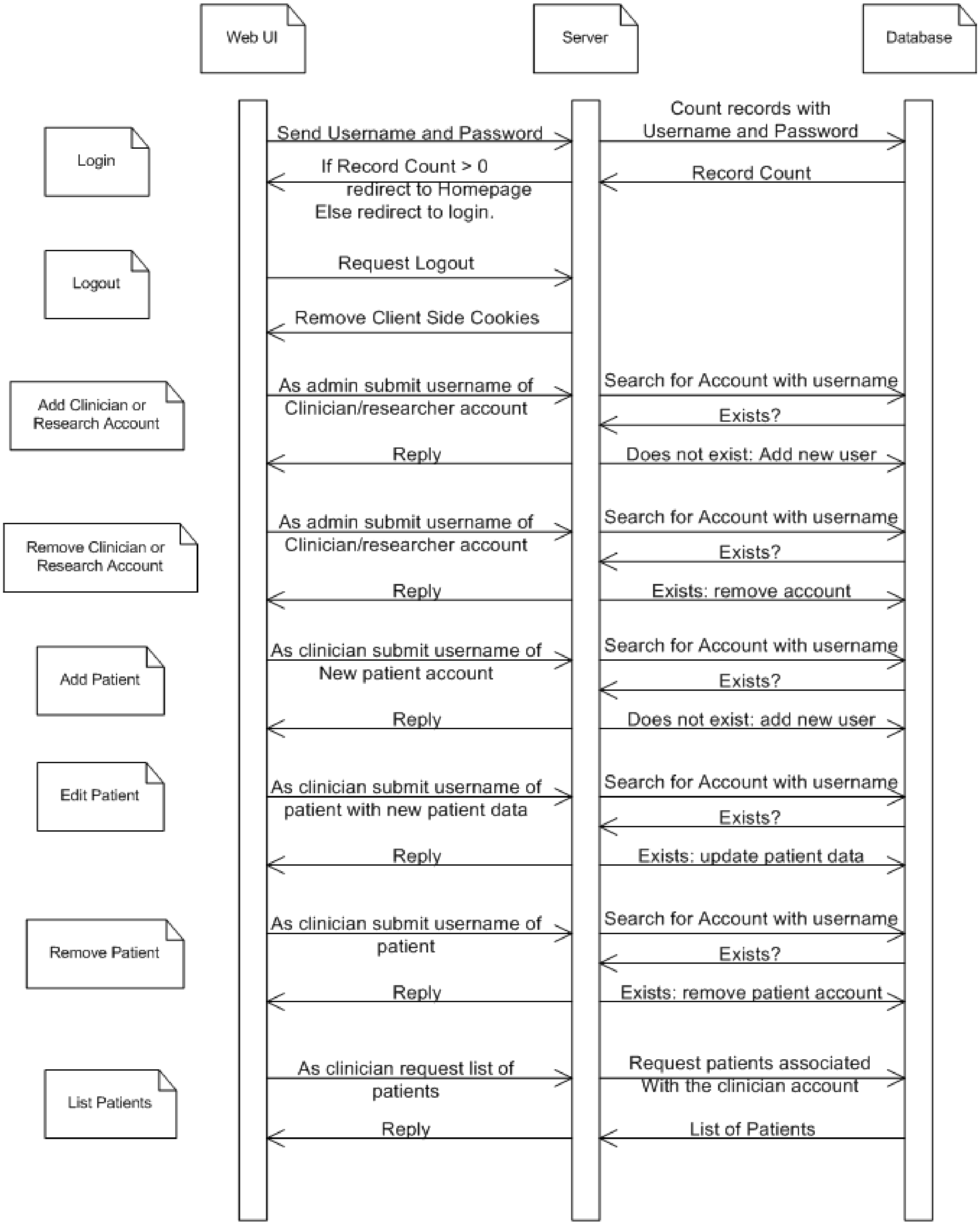
Trigger The data dump task is triggered by the completion of a day. The actual data dump request occurs after authentication with the Server.

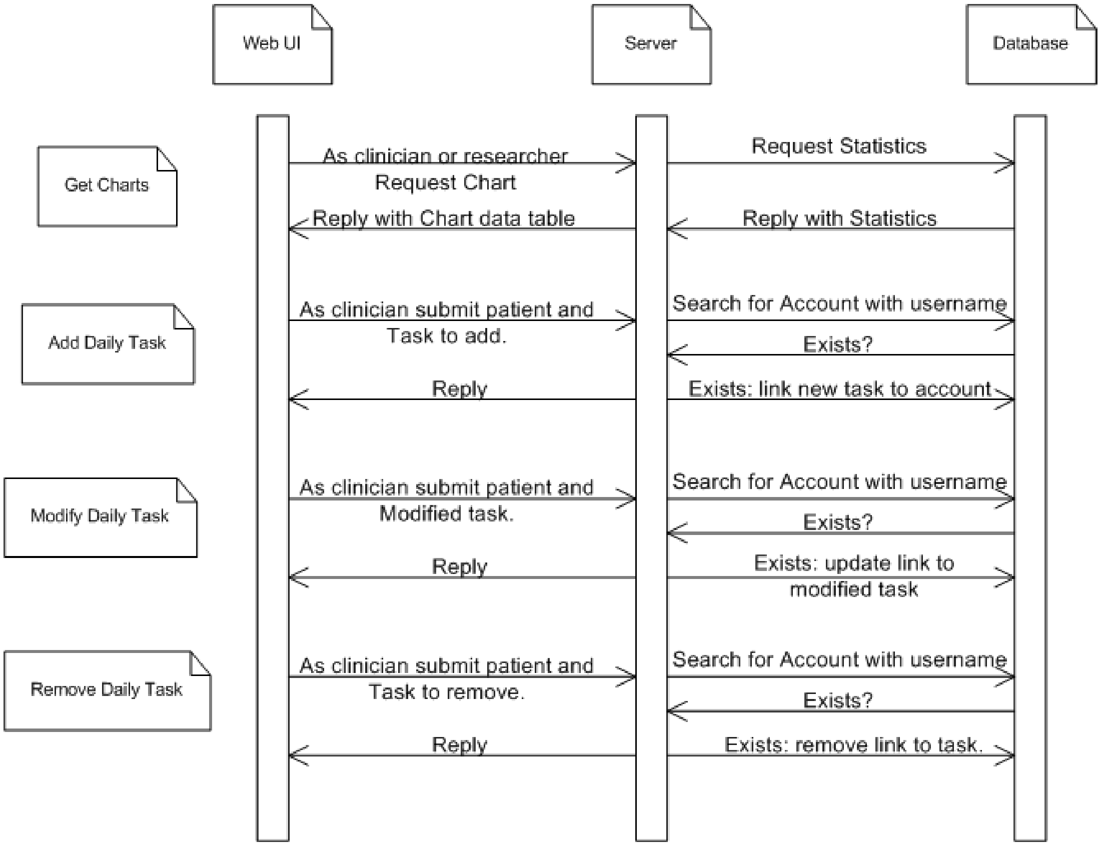
Request Daily Task

Description The user must be able to request a daily task from the database. At the beginning of the day, or while a data dump is occurring, the user sends the request to the server. When this request is received by the server, the server will search through the database to retrieve the tasks for the specified users. When the tasks are found, they are sent back to the PDA, allowing the user to view them.

Trigger The request daily task event occurs before the beginning of a new day or during a data dump connection. The actual request is sent after authentication with the Server.

9.2 Web UI Interfaces


Web UI
Server
Database
Login
Send Username and Password
Count records with Username and Password
If Record Count > 0 redirect to Homepage Else redirect to login.
Record Count
Logout
Request Logout
Remove Client Side Cookies
Add Clinician or Research Account
As admin submit username of Clinician/researcher account
Search for Account with username
Exists?
Reply
Does not exist: Add new user
Remove Clinician or Research Account
As admin submit username of Clinician/researcher account
Search for Account with username
Exists?
Reply
Exists: remove account
Add Patient
As clinician submit username of New patient account
Search for Account with username
Exists?
Reply
Does not exist: add new user
Edit Patient
As clinician submit username of patient with new patient data
Search for Account with username
Exists?
Reply
Exists: update patient data
Remove Patient
As clinician submit username of patient
Search for Account with username
Exists?
Reply
Exists: remove patient account
List Patients
As clinician request list of patients
Request patients associated With the clinician account
Reply
List of Patients


Web UI
Server
Database
Get Charts
As clinician or researcher Request Chart
Reply with Chart data table
Request Statistics
Reply with Statistics
Add Daily Task
As clinician submit patient and Task to add.
Reply
Search for Account with username
Exists?
Exists: link new task to account
Modify Daily Task
As clinician submit patient and Modified task.
Reply
Search for Account with username
Exists?
Exists: update link to modified task
Remove Daily Task
As clinician submit patient and Task to remove.
Reply
Search for Account with username
Exists?
Exists: remove link to task.

Login

Description The clinician or researcher is attempting to access the privileged material available within the Web UI. In order to access this material the clinician or researcher must be a registered user of the system.

Trigger A request is made to the Web UI to login an account with the provided username and password. Generally this request is generated by a web browser when the client clicks the “Login” button on the login web page.

Logout

Description The task should remove any client side cookies and data from the clients web browser. This will prevent the client from logging in again until authentication is provided.

Trigger A request is made to the Web UI to logout of a clients account. Generally, this request is generated by a web browser when the client clicks the “Logout” link.

Add Clinician or Research Account

Description The administrative user wishes to grant access to a new clinician or researcher. This task will add information describing the new clinician or researcher and create an account for the new user in the database.

Trigger The administrative user submits a request to the Web UI to add a new clinician or researcher account. The request includes the full name, privileges, address, and contact information for the new user. Generally, this request will be generated when the administrative user submits the add new user page of the Web UI.

Remove Clinician or Research Account

Description The administrative user wishes to prevent access of a current clinician or researcher to the Web UI. This is done by removing the clinicians or researchers account. The removal request includes the name of the account to be removed.

Trigger The administrative user submits a request to the Web UI to remove a clinician or researcher account. Generally, this request will be generated when the administrative user submits the remove user page of the Web UI.

Add Patient

Description A clinician is adding a new patient to the system. The request to add the patient includes only a username for the patient and a password.

Trigger A request to add a new patient is submitted by a logged in clinician. Generally, this request is generated and submitted when the clinician submits the add new user form.

Edit Patient

Description The clinician wants to edit the patients profile. By default for new patients, the profile is empty and must be updated at least once by the clinician. The edit profile request from the Web UI includes the username of the patient and the patients personal data.

Trigger A request is submitted to the Web UI to update a patient's profile. Generally this occurs when the clinician submits changes to the edit patient profile page.

Remove Patient

Description The patient is no longer participating in the Activity Monitor system and the clinician wants to remove them. The request to do this includes the patient's username.

Trigger A request is submitted to the Web UI to remove a patient's username from the system. Generally, this request is sent automatically when the clinician clicks the remove patient link on the patient's profile.

List Patients

Description The list patients task provides a list of all patients the currently logged in clinician is responsible for.

Trigger This event occurs when the clinician views the list patients page and the Web UI must request a list of all current patients linked to the clinician.

Get Charts

Description The clinician or researcher wants to view the progress of the patient(s) under their supervision. This includes the charts for the Group, Patient Completion, and Health. The Web UI request includes the group or patients which are part of the chart and the range of time the chart should be generated for along with the type of chart being generated.

Trigger A request for to view a specific chart is received by the Web UI. Generally, this request is generated by the clinician's or researcher's browser when the view chart page is accessed.

Add Daily Task

Description The clinician is adding a daily task for a specific patient. The request includes the username of the patient and the data representing the task to link to the patients data.

Trigger The Web UI receives a request to add a daily task for a patient. Generally, this request is sent from the clinician's browser when the clinician selects the add/edit daily task for patient on the patient's profile page.

Modify Daily Task

Description The clinician wants to change the daily task for the patient. This maybe necessary to advance the patient's rehabilitation efforts. The request includes the patient's username and the new task for the patient.

Trigger The Web UI receives a request to modify the daily task for a patient. Generally, this request is sent from the clinician's browser when the clinicians attempts to save the modified daily task on the add/edit daily task page.

Remove Daily Task

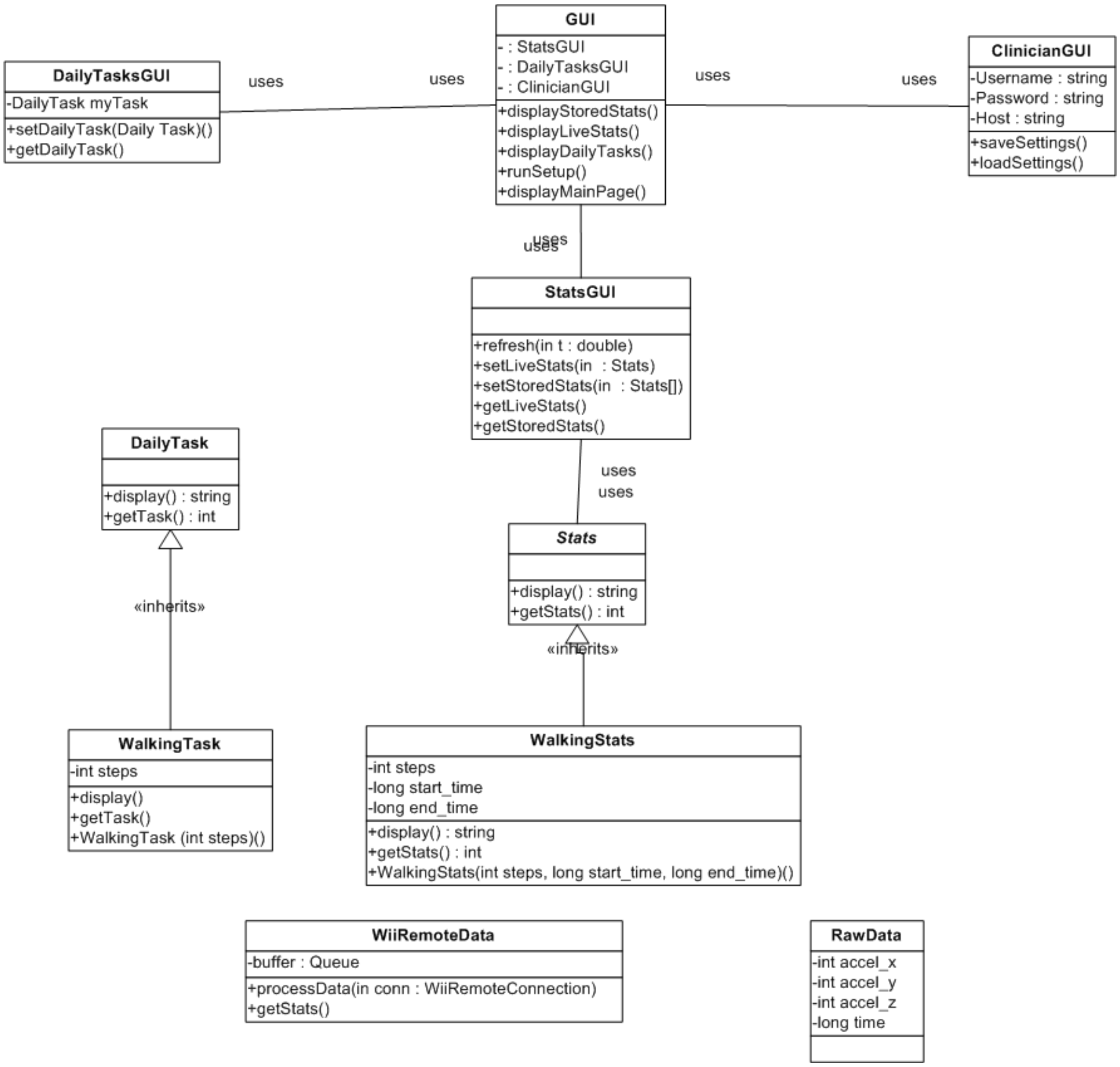
Description The clinician is going to remove the daily task for a patient. This means the patient will not have to fulfill any requirements, but instead will only have the activities he or she performs recorded. The request to remove the daily task includes the username of the patient.

Trigger The Web UI receives a request to remove the daily task for a patient. Generally, this request is sent from the clinician's web browser when the remove daily task link is selected from the patient's profile page.

10 Detailed Class Diagrams

10.1 PDA Class Diagrams

10.1.1 PDA Patient GUI



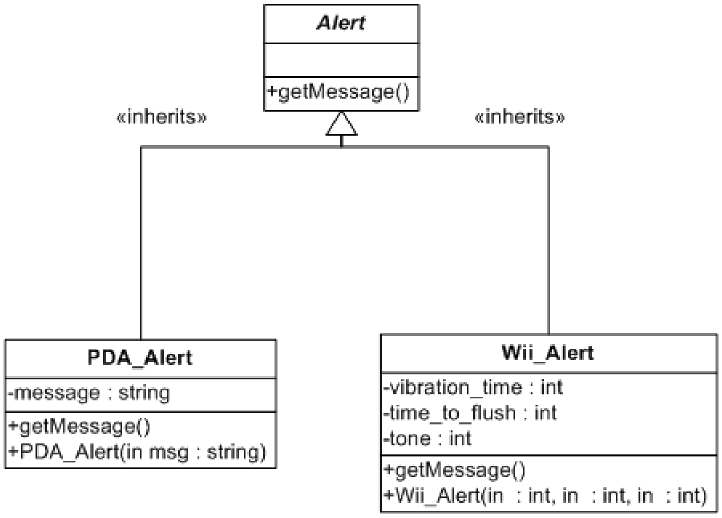
Class	Method	Description
GUI	displayStoredStats()	This will bring up the GUI that will display the statistics from previous sessions.
	displayLiveStats()	This will bring up the GUI that will display real time statistics that are currently being measured.
	displayDailyTasks()	This will bring up the GUI that will display the daily tasks.

	runSetup()	This will run the clinician setup and bring up the ClinicianGui so the clinician can enter the patients username, password and the host of the server's address.
	displayMainPage()	This will display the main page that the user will see when they start the application.

Stats	display()
	getStats()
		This is a pure virtual function that will display the PDA Statistics through an easy to read GUI.
		This is a pure virtual function that will retrieve the PDA Statistics.

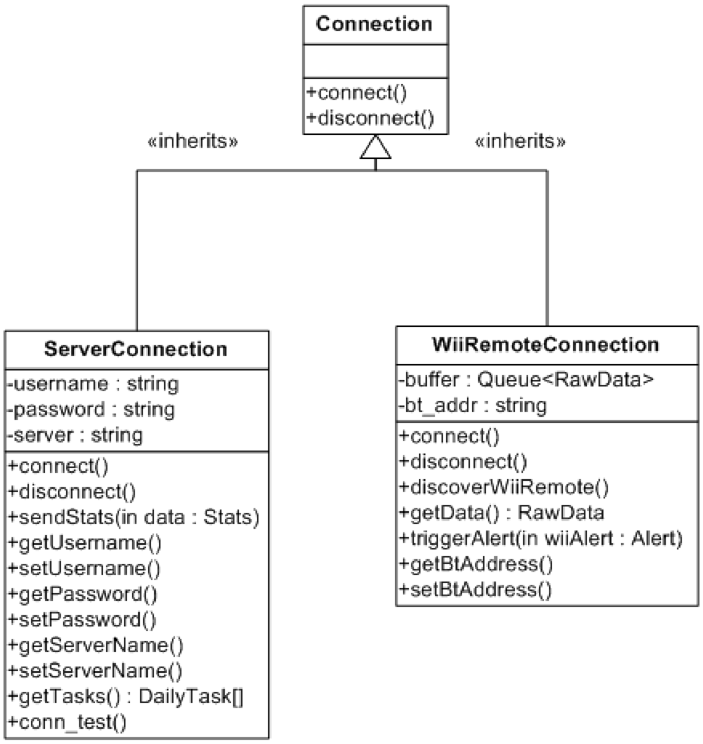
WalkingStats	display()	Virtual, derived function from the "Stats" abstract base. Displays the Walking Stats on the PDA through an easy to read GUI.
	getStats()	Retrieves the walking statistics to be displayed on the PDA.
	WalkingStats(int steps, long startTime, long endTime)	Constructor for the WalkingStats object that sets the object's attributes.
DailyTasksGui	setDailyTask(Daily Task)	This will set the Daily Task myTask with whatever is passed in to it.
	getDailyTask()	This will return an object of type Daily Task that has been stored in myTask.
WiiRemoteData	processData (WiiRemoteConnection)	Periodically get the data from the provided WiiRemoteConnection and determine where the steps occur.
	Stats getStats()	Return the next Stats object in the buffer. The Stats objects are created once enough raw data has been received.
ClinicianGUI	saveSettings()	Save the current settings to the configuration file.
	loadSettings()	Load the settings saved in the configuration file if it exists.
DailyTask	string display()	Provides a human readable sentence indicating the task represented.
	int *getTask()	Provides a pointer to either a single integer or an array of integer which can be used to represent the daily task.
WalkingTask	string display()	Provides a human readable sentence of the form: "Walk xxx steps".
	int* getTask()	Provides a pointer to a single integer which represents the number of steps to walk that day.
	WalkingTask(int steps)	Constructor which initializes the task to be walk the number of steps specified.

10.1.2 PDA Alert Hierarchy



Class	Method	Description
Alert	getMessage()	This will receive a message for an alert.
PDA_Alert	getMessage()	This will receive a message for an alert to the PDA.
	PDA_Alert(string)	This will alert the PDA with a certain message.
Wii_Alert	getMessage()	This will receive a message for an alert to the Wii Remote.
	Wii_Alert(int, int, int)	This will alert the Wii Remote with a certain vibration and flash time as well as tone.

10.1.3 PDA-Server Connection

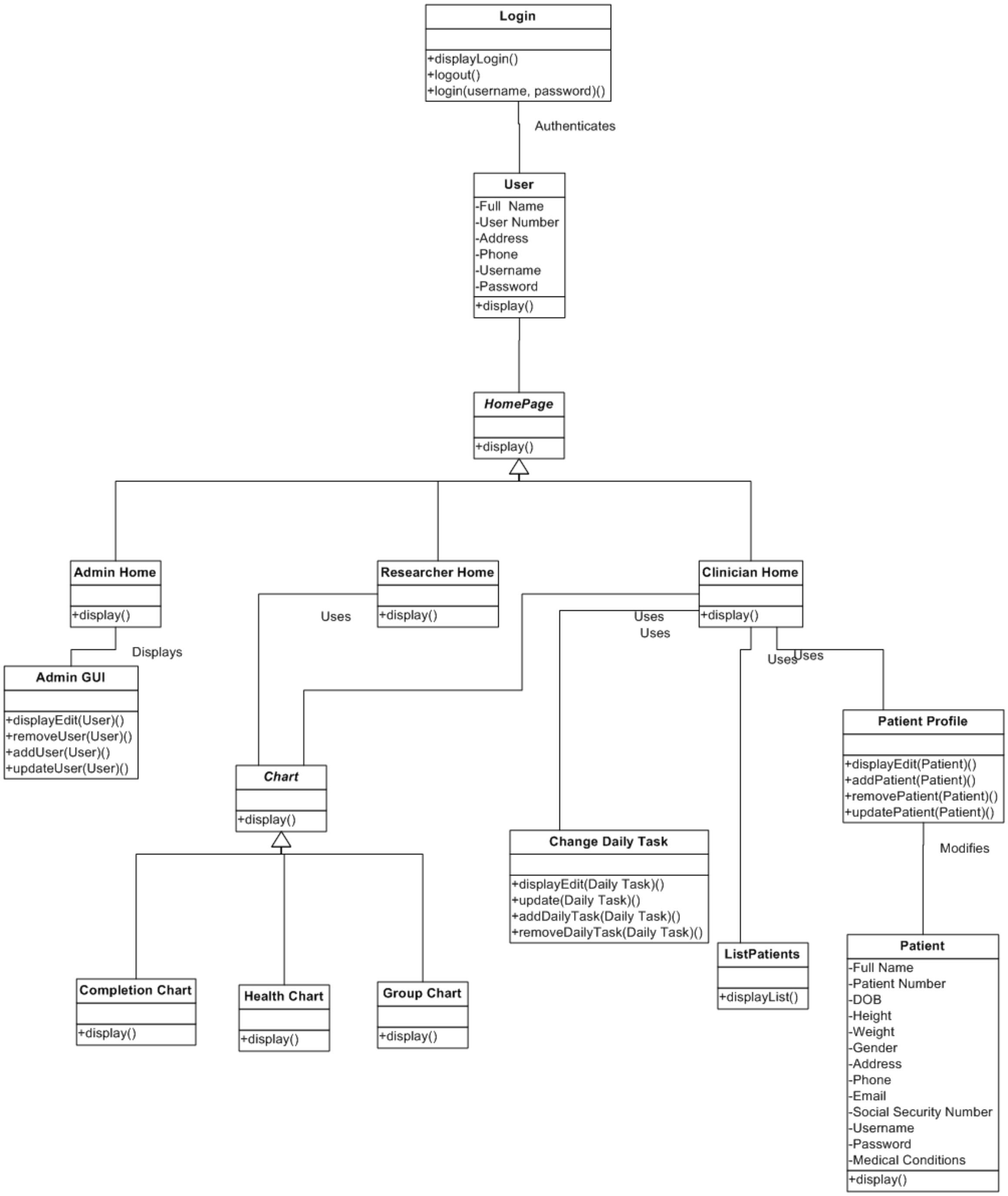


Class	Method	Description
Connection	connect()	This is a pure virtual function that will connect the PDA to the Server and Wii Remote respectively.
	disconnect()	This is a pure virtual function that will disconnect the PDA to the Server and Wii Remote respectively.
ServerConnection	connect()	Virtual, derived function from the "Connection" abstract base. Connects the PDA to the server.
	disconnect()	Virtual, derived function from the "Connection" abstract base. Disconnects the PDA from the server.
	sendStats()	Sends PDA statistics to the server.
	getUsername()	Retrieves the username that the PDA will use to connect to the server.
	getPassword()	Retrieves the password that the PDA will use to connect to the server.
	setUsername()	Sets the username on the server that the PDA will connect to.
	setPassword()	Sets the password to the server that the PDA will connect to.
	getServerName()	Retrieves the server name that the PDA is connected to.

	setServerName()	Sets the name of the server that the PDA will connect to.
	getTasks() : DailyTask[]	Retrieves daily tasks for patient from the server as an array.
	conn_test()	Runs a connection test to the server, returns status attempt to connect.
WiiRemoteConnection	connect()	Virtual, derived function from the “Connection” abstract base. Connects the PDA to the Wii Remote.
	disconnect()	Virtual, derived function from the “Connection” abstract base. Disconnects the PDA from the Wii Remote.
	discoverWiiRemote()	Function that looks for a Bluetooth Wii Remote that is specified by a Bluetooth address to be connected to the PDA.
	getData() : RawData	Retrieves data being sent by the Wii Remote and returns a RawData structure.
	triggerAlert(in wiiAlert:Alert)	This function takes an object of type Alert and will send the alert through to the PDA GUI.
	getBtAddress()	Retrieves the Bluetooth address of the Wii Remote connected to the PDA.
	setBtAddress()	Sets the bluetooth address of the Wii Remote to be connected to the PDA.

10.2 Server Class Diagrams

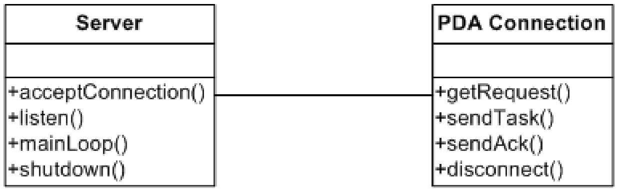
10.2.1 Server WebUI



Class	**Method**	**Description**
AdminGUI	displayEdit(User)	Generates an editable form for modifying the fields associated with the provided user. If the User.Username field is blank this form will create a new user. Otherwise it will autofill the fields with the values provided by the User object.
	removeUser(User)	Remove the user with User.Username from the database.
	addUser(User)	Add a user to the database with the data provided by the User object.
	updateUser(User)	Update the information in the database with the provided information for the user with username User.Username.
Login	displayLogin()	This will display the login page for all users.
	logout()	This will log the current user out.
	login(Username, Password)	This will take the populated fields for username and password, and return user type from the database if successful.
ListPatients	displayList()	This will give a list of all the clinician's patients on the interface.
PatientProfile	displayEdit(Patient)	This will take a Patient object as a parameter and display all the fields in editable form.
	addPatient(Patient)	This will take a Patient object as a parameter and add a new patient to the database.
	removePatient(Patient)	This will take a Patient object as a parameter and remove that patient from the database.
	updatePatient(Patient)	This will take a Patient object as a parameter and update an existing patient to the database.
Chart	display()	Virtual function that will display the derived charts after getting the required data from the database.
CompletionChart	display()	This function will display the patient completion charts by getting comparing patient statistics to the tasks that belong to them.
GroupChart	display()	This function will gather data from a Clinician's or Researcher's group and graph it.
HealthChart	display()	This function will get data that relates to the patients health and graph it.
Homepage	display()	This is a pure virtual function that will display the respective homepages after a specific user (Clinician, Researcher, or Admin) logs into the Web UI.

Admin Homepage	display()	Virtual, derived function from the “Homepage” abstract base. Displays the Admin’s Homepage which allows the Admin to Add/Remove/Edit users of the Web UI.
Researcher Homepage	display()	Virtual, derived function from the “Homepage” abstract base. Displays the Researcher’s Homepage which gives the Researcher access to the Researcher Charts.
Clinician Homepage	display()	Virtual, derived function from the “Homepage” abstract base. Displays the Clinician’s Homepage which allows the Clinician to Manage Patients, Change Daily Tasks, and View Clinician Charts.
Change Daily Task	displayEdit(Daily Task)	Shows the Daily Task specified as its argument in editable fields so that the Clinician can easily modify it.
	update(Daily Task)	Function that gets called to actually update the Daily Task after changes are made by the Clinician.
	addDailyTask(Daily Task)	Function that gets called when a new Daily Task is submitted by the Clinician. The function will take the Daily Task object and call the appropriate DAO to add it to the Database.
	removeDailyTask(Daily Task)	Function that gets called when a Daily Task is to be removed from a certain Patient’s Profile.

10.2.2 Server-PDA Connection



Class	Method	Description
Server	acceptConnection()	Accept a connection from PDA and create the PDAConnection object to associate with it.
	listen()	Starts the listening socket.
	mainLoop()	Called when initialized. This method opens a listening socket and waits for the PDA connections.
	shutdown()	Close the listening socket and shutdown the server.

PDAConnection	getRequest()	Receive requests from the PDA and parse them into upload Stats or request of Daily Task messages.
	sendTask()	Send the daily task to the PDA.
	sendAck()	Send a ACK or NACK to the PDA to indicate whether the Stats were successfully received or not.
	disconnect()	Break the connection with the associated PDA object.

11 Unit Test Plan

The convention for each method is to list the unique test identifier as the first paragraph. The unique test identifier is generated by taking the initials of the object subscript the initials of the method.

11.1 PDA Patient GUI

11.1.1 GUI Object

displayStoredStats()

GUI$_{dss}$

Risk Analysis This is a low risk system component due to the fact that it will just start up the StatsGUI function getStoredStats().

Possible Defects If getStoredStats is not completed it will fail.

Completion Criteria For this object to complete sucessfuly it will call the StatsGui function getStoredStats().

displayLiveStats()

GUI$_{dls}$

Risk Analysis This is a low risk system component due to the fact that it will just start up the StatsGUI function getLiveStats().

Possible Defects If getLiveStats is not completed it will fail.

Completion Criteria For this object to complete sucessfuly it will call the StatsGui function getLiveStats().

displayDailyTasks()

GUI$_{ddt}$

Risk Analysis This is a low risk system component due to the fact that it will just start up the DailyTasksGUI function getDailyTask().

Possible Defects If getDailyTask is not completed it will fail.

Completion Criteria For this object to complete sucessfuly it will call the DailyTasksGUI function getDailyTask().

runSetup()

GUI$_{rs}$

Risk Analysis This is a low risk system component due to the fact that it will just start up the ClinicianGUI.

Possible Defects If the ClinicianGUI is not completed it will fail.

Completion Criteria For this object to complete sucessfuly it will start the ClinicianGUI.

displayMainPage()

GUI$_{dmp}$

Risk Analysis This is a high risk system component due to it being the GUI from which all the navigation starts, and also returns to.

Possible Defects The GUI not going to other GUI pages.
The GUI not displaying all options.

Completion Criteria For this to be complete it must show all the options and travle to all the other GUI pages.

11.1.2 ClinicianGUI

saveSettings()

CGUI$_{ss}$

Risk Analysis This is a medium risk method. It is medium risk because it is responsible for file input and output with the configuration file.

Possible Defects

1. A defect would be if the method wrote the wrong or incomplete settings to the configuration file.
2. Another defect would be if the settings were written in an incorrect format.
3. A third defect would be if the file was not closed properly preventing read/write access from other processes on the system.

Completion Criteria The function will be considered complete when it is capable of writing the settings to the configuration file and closing the file correctly. It must write the settings in the correct format so the settings can be loaded from the configuration file.

loadSettings()

CGUI$_{ls}$

Risk Analysis This is a medium risk to implement function because it requires file io. The function must retrieve the saved settings from the configuration file.

Possible Defects

1. The method incorrectly reads the format of the file and therefore the incorrect settings are loaded.
2. The file is incorrectly formatted to begin with and the function unknowingly loads the erroneous settings.
3. The file is not closed correctly once the settings are read into the system.

Completion Criteria The function will be considered complete when it correctly retrieves the settings from the configuration file and closes the file when finished.

11.1.3 DailyTasksGui Object

setDailyTask(Daily Task)

DTG$_{sdt}$

Risk Analysis This is a low risk system component because it just grabs a copy of daily task from the DailyTask object.

Possible Defects If DailyTask has not been completed it may not return an object to DailyTasksGui.

Completion Criteria This function needs DailyTask getTask() to return a DailyTask and it will store it in DailyTasksGUI.

getDailyTask()

DTG$_{gdt}$

Risk Analysis This is a meduim risk system component that will take the stored DailyTask and display it to the screen.

Possible Defects If setDailyTask does not work then there will be no task to display.

Completion Criteria The Daily Task needs to be displayed and it must let the user navigate back to the main page.

11.1.4 WiiRemoteData Object

processData(WiiRemoteConnection)

WRD$_{pd}$

Risk Analysis This is a high risk system component due to the constant need of reading in and processing new data. This could lead to memory leaks or delays in processing may crop up if the system is unable to process the data fast enough.

Possible Defects

1. The function is unable to handle the Wii Remote sampling rate and a backlog or RawData objects occurs. This would slow down system performance.
2. The function does not produce the correct Statistics objects when provided with predetermined data.

Completion Criteria This method will be considered complete when it is capable of handling input data and producing the correct Stats objects for the WiiRemoteData internal buffer. The data must be processed in real time and the amount of memory used by this component should not increase over time.

Stats getStats()

WRD$_{gs}$

Risk Analysis This is a low risk method because of it is a get method and only responsible for taking the next Stats object off the internal buffer.

Possible Defects If the function does not remove the object it removes from the front of the queue it is considered a defect.

Completion Criteria The method will be considered complete when it returns and removes the first Stats object from the internal queue.

11.1.5 DailyTask

Since this is an abstract data type it does not require a test plan.

11.1.6 WalkingTask

display()

WT$_{d}$

Risk Analysis This is a low risk function. The function is low risk, because it is responsible for taking the internal values of the object and creating a human readable sentence.

Possible Defects If the function does not create a human readable sentence with the internal walking data then it is considered a defect.

Completion Criteria The function will be complete when it creates a human readable sentence which can be used for display purposes that represents the internal data of the object.

getStats()

WT$_{gs}$

Risk Analysis This is a low risk method. It is considered low risk because it is a get method and involves no serious data processing.

Possible Defects If the method returns a pointer to an array of integers located on the stack instead of on the heap it will be considered defective.

Completion Criteria The method will be considered complete when it returns a pointer to a heap allocated array of integers representing the internal data of the object.

WalkingTask(int steps)

WT$_{wt}$

Risk Analysis This is a low risk method, because it is responsible for setting the internal data member of the object. There is no data processing or allocation of resources involved.

Possible Defects If the internal data member is not set to the provided value the method is defective.

Completion Criteria If the method takes an input value and sets the internal data member of the object to the input then the method will be considered completed.

11.1.7 Stats Object

display()

ST$_{d}$

Risk Analysis This is a low risk system component due to the fact that it will just call the display function from a specific Stats object.

Possible Defects The display function is not called.

Completion Criteria The stats are displayed.

getStats()

ST$_{gs}$

Risk Analysis This is a low risk system component due to the fact that it will just call the getStats() function from a specific Stats object.

Possible Defects The statistics are not retrieved.

Completion Criteria The statistics are retrieved.

11.1.8 WalkingStats Object

display()

WST$_d$

Risk Analysis This is a low risk system component due to the fact that it will just display the task attributes.

Possible Defects The Walking Task is incorrectly displayed.

Completion Criteria The Walking Task is displayed correctly.

getStats()

WST$_{gs}$

Risk Analysis This is a low risk system component due to the fact that it will just retrieve the attributes stored in the object.

Possible Defects The attributes are not returned.

Completion Criteria The task attributes are retrieved.

WalkingTask(int steps, long start_time, long end_time)

WST$_{wt}$

Risk Analysis This is a low risk system component due to the fact that it will just set the objects attributes.

Possible Defects The attributes are not set correctly.

Completion Criteria The object's attributes are set correctly.

11.2 PDA Alert Hierarchy

11.2.1 Alert Object

getMessage()

Alert$_{gm}$

Risk Analysis This has a low risk factor since it just receives a message for an alert.

Possible Defects No alert will occur without a message being received.

Completion Criteria For the completion to be successful it will receive a message for the alert to be called.

11.2.2 PDAAlert Object

getMessage()

PDAAlert$_{gm}$

Risk Analysis This has a low risk factor since it just receives a message for an alert for the PDA.

Possible Defects No alert will occur without a message being received for the PDA.

Completion Criteria For the completion to be successful it will receive a message for the alert to be called on the PDA.

PDAAlert(string)

PDAAlert$_{pa}$

Risk Analysis This has a low risk factor since it just alerts the PDA with a message.

Possible Defects No alert will occur if there isn't a message for the parameter.

Completion Criteria For the completion to be successful it will display an alert on the PDA with a certain message.

11.2.3 WiiAlert Object

getMessage()

WiiAlert$_{gm}$

Risk Analysis This has a low risk factor since it just receives a message for an alert for the Wii Remote.

Possible Defects No alert will occur without a message being received for the Wii Remote.

Completion Criteria For the completion to be successful it will receive a message for the alert on the Wii Remote.

WiiAlert(int,int,int)

WiiAlert$_{wa}$

Risk Analysis This has a medium risk factor because it alerts the Wii Remote with vibrations and flashes for notification.

Possible Defects No alert will occur and the client will not be notified for any progress.

Completion Criteria For the completion to be successful it will alert the Wii Remote with vibration and flash times.

11.3 PDA-Server Connection

11.3.1 Connection

connect()

Connection$_{con}$

Risk Analysis This is a low risk system component due to the fact that it will just call the appropriate connect() function in either the WiiRemoteConnection or ServerConnection object.

Possible Defects If connect is not completed the system may not have all of its compnents connected and will fail.

Completion Criteria For this object to complete sucessfuly it will call the appropriate connect() function in either the WiiRemoteConnection or ServerConnection object.

disconnect()

Connection$_{dis}$

Risk Analysis This is a low risk system component due to the fact that it will just call the appropriate disconnect() function in either the WiiRemoteConnection or ServerConnection object.

Possible Defects If disconnect is not completed the system may have components still connected when they should not be.

Completion Criteria For this object to complete sucessfuly it will call the appropriate disconnect() function in either the WiiRemoteConnection or ServerConnection object.

11.3.2 ServerConnection Object

connect()

SC$_{conn}$

Risk Analysis This is a high risk system component because it is the method for which the PDA connects to the Server.

Possible Defects If connect() has not been completed the data flow between the PDA and server cannot be completed or started.

Completion Criteria The PDA connects to the server.

disconnect()

SC$_{dis}$

Risk Analysis This is a low risk system component that will disconnect the PDA from the server.

Possible Defects If PDA does not disconnect, will not impact system.

Completion Criteria The PDA is disconnected from the server.

sendStats()

SC$_{ss}$

Risk Analysis This is a meduim risk system component that will send the stored stats on the PDA to the server.

Possible Defects If sendStats() does not work, the server will not be able to collect patient data.

Completion Criteria The Server needs to receive stats sent by the PDA.

getUsername()

SC$_{gun}$

Risk Analysis This is a medium risk system component that will retrieve the username the PDA needs to connect to the server.

Possible Defects If getUsername does not work the PDA will unable to use the correct credentials to connect to the server.

Completion Criteria The PDA retrieves the username for the server.

getPassword()

SC$_{gpw}$

Risk Analysis This is a meduim risk system component that will retrieve the password the PDA needs to connect to the server.

Possible Defects If getPassword does not work the PDA will unable to use the correct credentials to connect to the server.

Completion Criteria The PDA retrieves the password for the server.

setUsername()

SC$_{sun}$

Risk Analysis This is a meduim risk system component that will set the username that the PDA uses to connect to the server.

Possible Defects If setUsername() does not work, the username required to connect to the server may not be updated.

Completion Criteria The username stored on the PDA to connect to the server is changed.

setPassword()

SC$_{spw}$

Risk Analysis This is a meduim risk system component that will set the password that the PDA uses to connect to the server.

Possible Defects If setPassword() does not work, the username required to connect to the server may not be updated.

Completion Criteria The password stored on the PDA to connect to the server is changed.

getServerName()

SC$_{gsn}$

Risk Analysis This is a medium risk system component that will retrieve the Server Name the PDA needs to connect to the server.

Possible Defects If getServerName does not work the PDA will unable to use the correct credentials to connect to the server.

Completion Criteria The PDA retrieves the Server Name for the server.

setServerName()

SC$_{ssn}$

Risk Analysis This is a meduim risk system component that will set the Server Name that the PDA uses to connect to the server.

Possible Defects If setServerName() does not work, the Server Name required to connect to the server may not be updated.

Completion Criteria The Server Name stored on the PDA to connect to the server is changed.

getTasks()

SC$_{gt}$

Risk Analysis This is a high risk system component that will retireve an array of daily tasks from the server.

Possible Defects If getTasks does not work then there will be no tasks on the PDA.

Completion Criteria The PDA is able to display new daily tasks retrieved from the server.

conn_test()

SC$_{connt}$

Risk Analysis This is a meduim risk system component that will test the connection to the Server.

Possible Defects If conn_test() fails, then there will be no way to tell if the connection to the server is valid.

Completion Criteria The conn_test() function returns the correct status of the Server to PDA connection.

11.3.3 WiiRemoteConnection Object

connect()

WRC$_{conn}$

Risk Analysis This is a high risk system component because it is the method for which the PDA connects to the Wii Remote.

Possible Defects If connect() has not been completed the data flow between the PDA and Wii Remote cannot be completed or started.

Completion Criteria The PDA connects to the Wii Remote.

disconnect()

WRC$_{dis}$

Risk Analysis This is a low risk system component that will disconnect the PDA from the Wii Remote.

Possible Defects If PDA does not disconnect, will not impact system.

Completion Criteria The PDA is disconnected from the Wii Remote.

discoverWiiRemote()

WRC$_{dwr}$

Risk Analysis This is a high risk system component as the PDA needs to discover the Wii Remote to connect to it.

Possible Defects The PDA is incapable of discovering Blue tooth Wii Remote .

Completion Criteria PDA recognizes Wii Remote device.

getData()

WRC$_{gd}$

Risk Analysis This is a high risk system component as processing Wii Remote data is the main function of the system.

Possible Defects Wii Remote data is not returned.

Completion Criteria Wii Remote data is received on the PDA and able to be processed.

triggerAlert(Alert)

WRC$_{tra}$

Risk Analysis This is a low risk system component as alerts are not necessary for core system functionailty of processing data.

Possible Defects Alert is not triggered when an event that should trigger one occurs.

Completion Criteria PDA alert appears to patient when appropriate event occurs.

getBtAddress()

WRC$_{gbta}$

Risk Analysis This is a medium risk system component since retrieving the Bluetooth address of the Wii Remote is necessary for proper connection.

Possible Defects PDA unable to determine address of Wii Remote.

Completion Criteria PDA recognizes bluetooth address of Wii Remote.

setBtAddress()

WRC$_{sbta}$

Risk Analysis This is a medium risk system component since the result of setting the bluetooth address impacts the PDA's ability to connect to the Wii Remote.

Possible Defects PDA does not overwrite bluetooth address stored on PDA, unable to pair device.

Completion Criteria Bluetooth address for Wii Remote updated on PDA.

11.4 Server WebUI

ListPatients()

GUI$_{lp}$

Risk Analysis This has a medium risk factor because it displays a list of all a client's patients.

Possible Defects No list of patients will be displayed and further analysis can not be implemented.

Completion Criteria For the completion to be successful it will display a list of all a client's patients on the UI.

11.4.1 Login Object

displayLogin()

Login$_{dl}$

Risk Analysis This has a medium risk factor since it just displays the login page.

Possible Defects The login page will not display and nobody can login to their account.

Completion Criteria For the completion to be successful it display the login page for all users.

logout()

Login$_{lo}$

Risk Analysis This has a low risk factor since it just signs a user out of his/her account.

Possible Defects The user will not be able to logout of their account.

Completion Criteria For the completion to be successful it will end a current user's session.

login(Username,Password)

Login$_{li}$

Risk Analysis This has a medium risk factor because it instiantiates a session for a user on the UI.

Possible Defects No user will be able to log into the system.

Completion Criteria For the completion to be successful a user will be able to log into the system and return user type from the database.

11.4.2 PatientProfile Object

displayEdit(Patient)

PP$_{de}$

Risk Analysis This has a medium risk factor because it displays edit for patient profiles.

Possible Defects The user is not able to see patients' credentials in editable form.

Completion Criteria For the completion to be successful it populates the UI with the patient profile in editable form.

addPatient(Patient)

PP$_{ap}$

Risk Analysis This has a medium risk factor because it allows addition of patients.

Possible Defects Patients can not be added into the database.

Completion Criteria For the completion to be successful a patient is securely added into the back-end.

removePatient(Patient)

PP$_{rp}$

Risk Analysis This has a low risk factor because it just deletes unused patients from the database.

Possible Defects Patients can not be deleted once added to the database.

Completion Criteria For the completion to be successful it removes a patient completely from the back-end.

updatePatient(Patient)

PP$_{de}$

Risk Analysis This has a medium risk factor because it allows edit for patient profiles.

Possible Defects Patients can not be edited once added to the database.

Completion Criteria For the completion to be successful a patient can be modified after being added to the back-end.

11.4.3 AdminGUI

displayEdit(User)

AGUI$_{de}$

Risk Analysis This is a low risk method because its primary purpose is to generate html code to display to the user. The html code should produce the edit user interface for the Admin user.

Possible Defects

1. If the page is not displayed as the GUI mockup indicates.
2. The page does not contain the initial values of all the fields when editing an existing user.

Completion Criteria The function will be considered complete when the output is a syntactically correct html string which when shown in a browser displays the edit user gui as drawn in the GUI mockups.

removeUser(User)

AGUI$_{ru}$

Risk Analysis This is a medium risk function. The function is medium risk because it must remove a user from the database.

Possible Defects

1. The user is not removed from the database.
2. The wrong user is removed from the database, or unnecessary user data is removed from the database.

Completion Criteria The function will be considered complete when it removes only the user specified from the database.

addUser(User)

AGUI$_{au}$

Risk Analysis This is a low risk function because it is inserting a record into the database.

Possible Defects

1. The wrong user object is created.
2. The user object is created when a user of that username already exists.

Completion Criteria The function will be considered complete when a new user record is created in the database for a new unique User object that has been specified. If the username is already in use the function should return an error.

updateUser(User)

AGUI$_{uu}$

Risk Analysis This is a medium risk function due to the need to update a record in the database.

Possible Defects

1. The wrong record is updated, or no record is updated at all.
2. If no record is updated and an error message is not returned.

Completion Criteria The function will be considered complete when the User object is successfully propagated onto the database. In other words, the correct record is updated with the correct values. If an error occurs, the function should return the appropriate error code.

11.4.4 Chart Object

display()

$\mathbf{C}_d$

Risk Analysis This is a low risk system component due to the the fact that nothing relies on the charts to complete.

Possible Defects Since this is a virtual function there should be no defects.

Completion Criteria Since this is a virtual function there should be no test for completion.

11.4.5 CompletonChart Object

display()

$\mathbf{CC}_d$

Risk Analysis This is a low risk system component due to the the fact that nothing relies on the charts to complete.

Possible Defects The chart might not display or might not display the correct information.

Completion Criteria The chart displays the correctly.

11.4.6 GroupChart Object

display()

$\mathbf{GC}_d$

Risk Analysis This is a low risk system component due to the the fact that nothing relies on the charts to complete.

Possible Defects The chart might not display or might not display the correct information.

Completion Criteria The chart displays the correctly.

11.4.7 HealthChart Object

display()

$\mathbf{HC}_d$

Risk Analysis This is a low risk system component due to the the fact that nothing relies on the charts to complete.

Possible Defects The chart might not display or might not display the correct information.

Completion Criteria The chart displays the correctly.

11.4.8 Admin HomePage

display()

$\mathbf{AH}_d$

Risk Analysis This is a medium risk system component. Researchers and Clinicians can always be added to the database manually, but a GUI is desirable.

Possible Defects The wrong homepage may be displayed. For example, a Clinician logs in and is brought to the Admin Homepage.

Completion Criteria The Admin logs into the Web UI through the login screen. The Admin is presented with a GUI representing their homepage. From here, the Admin can add/remove/edit Clinicians and Researchers.

11.4.9 Clinician HomePage

display()

$\mathbf{CH}_d$

Risk Analysis This is a high risk component. This module must be implemented correctly in order for the Clinician to inteface with the application and access core system functionality.

Possible Defects

1. The wrong homepage may be displayed
2. The homepage may not include all required functionality: Patient List, Add Patient, Modify Daily Tasks and Logout

Completion Criteria The Clinician logs into the Web UI through the login screen. The Clinician is presented with a GUI representing their homepage. The Clinician is presented with the toolbar on the left-hand side that allows the Clinician to access their Patient List, Add/Edit Patients (which includes the Modification of the Daily Tasks), and Logout.

11.4.10 Researcher HomePage

display()

RH$_d$

Risk Analysis This is a medium risk component. This module must be implemented correctly in order for the Researcher to inteface with the application to access Researcher Charts.

Possible Defects

1. The wrong homepage may be displayed
2. The homepage may not include all required functionality: View Group Patient Charts, Completion Charts, and Health Charts.

Completion Criteria The Researcher logs into the Web UI through the login screen. The Researcher is presented with a GUI representing their homepage. The Researcher is presented with the toolbar on the left-hand side that allows the Researcher to access all of the Researcher Charts (Completion, Health, and Group).

11.4.11 Change Daily Task

displayEdit(Daily Task)

CDT$_{de}$

Risk Analysis This is a high risk component. This module must be implemented correctly in order for the Clinician assign Daily Tasks to a Patient.

Possible Defects The wrong Daily Task may be displayed.

Completion Criteria This method will be considered complete when the Clinician is able to edit a Daily Task from the Patient Profile screen. The Edit Daily Task screen will allow the Clinician to edit the number of steps the Patient is supposed to walk.

update(Daily Task)

CDT$_u$

Risk Analysis This is a high risk component. This module must be implemented correctly in order for the Clinician assign Daily Tasks to a Patient.

Possible Defects

1. The wrong Daily Task gets updated.
2. Multiple Daily Tasks get updated.
3. The correct Daily Task gets updated with the wrong value(s).

Completion Criteria This method will be considered complete when the the correct Daily Task is actually updated in the database after the Clinician modifies it in the GUI.

addDailyTask(Daily Task)

CDT$_{adt}$

Risk Analysis This is a high risk component. This module must be implemented correctly in order for the Clinician assign Daily Tasks to a Patient.

Possible Defects The wrong Daily Task gets added

Completion Criteria This method will be considered complete when the the correct Daily Task is actually added into the database after the Clinician adds it via the Web UI.

removeDailyTask(Daily Task)

CDT$_{rdt}$

Risk Analysis This is a high risk component. This module must be implemented correctly in order for the Clinician remove Daily Tasks to a Patient.

Possible Defects

1. The wrong Daily Task gets removed from the database.
2. Multiple Daily Tasks get removed from the database.

Completion Criteria This method will be considered complete when the the correct Daily Task is actually removed into the database after the Clinician removes it via the Web UI.

11.5 Server-PDA Connection

11.5.1 Server

acceptConnection()

S$_{ac}$

Risk Analysis This is a low risk component as an accept connection routine is common to many servers already in existence. Therefore the design of this function is known.

Possible Defects

1. A possible defect is if the server did not accept the connection and allocate the resources required by the new connection. This allocation must occur in order to proceed.
2. Another defect would be if the server refused to accept connections.

Completion Criteria This item will be considered complete when it successfully accepts an incoming connection from a PDA and instantiates the correct objects to handle the remainder of the connection processing. It must also successfully pass the accepted client Socket connection off to the newly instantiated objects.

listen()

S$_l$

Risk Analysis This is another low risk function due to its common nature among servers. As long as the standard server model is followed there should be minimal risk.

Possible Defects One defect would be if the server did not open a secure port of listening.

Completion Criteria This method will be considered complete when a secure port on the server machine has been opened and the necessary parameters have been passed to the acceptConnection function or saved in the object.

mainLoop()

S$_{ml}$

Risk Analysis This is another low risk function. Since it is only responsible for controlling the server and ensuring user commands are executed it, there are few opportunities for issues to arise.

Possible Defects

1. A possible defect is if the function is unable to parse user input correctly.
2. Another defect would be if the input prompt freezes when attempting to perform an operation on the server.
3. The loop must not exist unless the user specifically requests the loop to terminate.

Completion Criteria The function will be considered complete when the user is able to enter control commands and the commands are correctly distributed to the appropriate function within the server.

11.5.2 PDAConnections

getRequest()

PDAC$_{gr}$

Risk Analysis This is a medium risk function as it is responsible for receiving all requests from the PDA and parsing them appropriately. This is reliant upon the successfully completion of a message protocol which will be used by both the PDA application and this function.

Possible Defects

1. Defects include losing complete requests which follow an incomplete request.
2. Also allocating resources for an incomplete or incorrect request is considered a defect.
3. A third defect is if the function does not exist gracefully when the socket layer connection is terminated.

Completion Criteria The function will be considered complete when it successfully receives a request from the PDA and successfully passes it to the appropriate request handler. If an erroneous request is received it should be ignore and the function should continue. When the socket connection is terminated the function should exit gracefully.

sendTask()

PDAC$_{st}$

Risk Analysis This is a medium risk method to implement. It must be able to respond to a send task request from the PDA using the appropriate message format. It must be able to successfully translate from the Stats object to a message.

Possible Defects A possible defect would be if the object was incorrectly translated to a message.

Completion Criteria It will be complete when it is capable of sending a Stats object over the PDA socket connection using the appropriate message format.

sendAck(boolean ack)

PDAC$_{sa}$

Risk Analysis This is a low risk method, because it must respond to the PDA with a ACK message or a NACK message.

Possible Defects If the messages do not follow the prescribed protocol, this is considered a defect.

Completion Criteria The task will be considered complete when the functions ends a properly formated ACK or NACK message depending upon the input parameter. An input of true should send an ACK message, while an input of false should send a NACK message.

disconnect()

PDAC$_{d}$

Risk Analysis This is a low risk function, due to its simple nature of terminating the connection between the Server and PDA.

Possible Defects A possible defect would be if the function did not terminate the connection and deallocate any resources used by the connection.

Completion Criteria The method will be considered complete when it terminates the connection between the PDA and the Server and deallocates all resources used by the connection.