MIS6308

System Analysis and Project Management

HealthReflex

Health Regulatory Smart Watch

Project Report

PROPERTY OF APURVA GODGHASE

Table of Contents

PROPERTY OF APURVA GODGHASE

Executive Summary

In today's day and age, data drives everything. Data is being gathered through traditional and non-traditional means to better understand human behavior and further utilized to make lives easier. Every traditional appliance or gadget can be thus exploited to collect data, forming an internet of things.

This project highlights one such application of data collection and use, that would lead to an enhancement to the lives of people who are aged, or who suffer from medical problems that require attention and support. **HealthReflex - Health Regulatory Smart Watch** is a wearable device, much like a watch that assists users in monitoring their health, setting up reminders and notifying contacts in case of emergency situations. It does this by leveraging the power of predictive analytics on crowd sourced and wearable collected data, detecting any emergencies or anomalies in regular functions and notifying emergency contacts to enable swift action. This includes detection of cardiopulmonary arrest through continuous tracking of the user's pulse. Location detection services are also provided for emergency contacts of people suffering from amnesia or neurodegenerative diseases like dementia or Alzheimer's disease. A panic button functionality enables users to reach out to their emergency contacts in times of distress.

The smart watch remains connected to the internet either through Wi-Fi or a smartphone with 3G/LTE connectivity. It makes use of a mobile application deployed on major mobile operating systems like iOS and Android to help the users in setup and inputting information, as well as toggling functionality according to each users' medical needs. Upon collection of initial data, the watch continuously interacts with other similar wearable devices, collecting and exchanging data. This revolutionary wearable device thus takes the concept of smart watches and applies it to the specific area of medical care and monitoring, to provide users and their families with some much-needed comfort and security in their day to day lives.

PROPERTY OF APURVA GODGHASE

Problem Statement

It is essential for people to monitor their health as they get older, or when suffering from ailments that require constant attention and care. Current systems include regular frequent doctor's visits and use of traditional gadgets to help gather data about an individual's health. This requires significant effort and time on the part of the individual and his family. This Health Regulatory Smart Watch addresses this need by providing individuals with a wearable smart device that constantly monitors their health, sends them medicine and appointment reminders, notifies in case of any abnormal medical statistics based on prior data and tracks user location.

HealthReflex thus addresses some of the major concerns faced by users and families in terms of medical care needs. These are

- Continuous tracking of user vitals; using predictive analytics to predict cardiac arrest or other anomalies, and notifying emergency contacts
- Reminders for medical appointments, follow-ups and medicine intake
- Inbuilt panic button functionality
- Location Tracking, especially for people suffering from amnesia or other neurodegenerative diseases

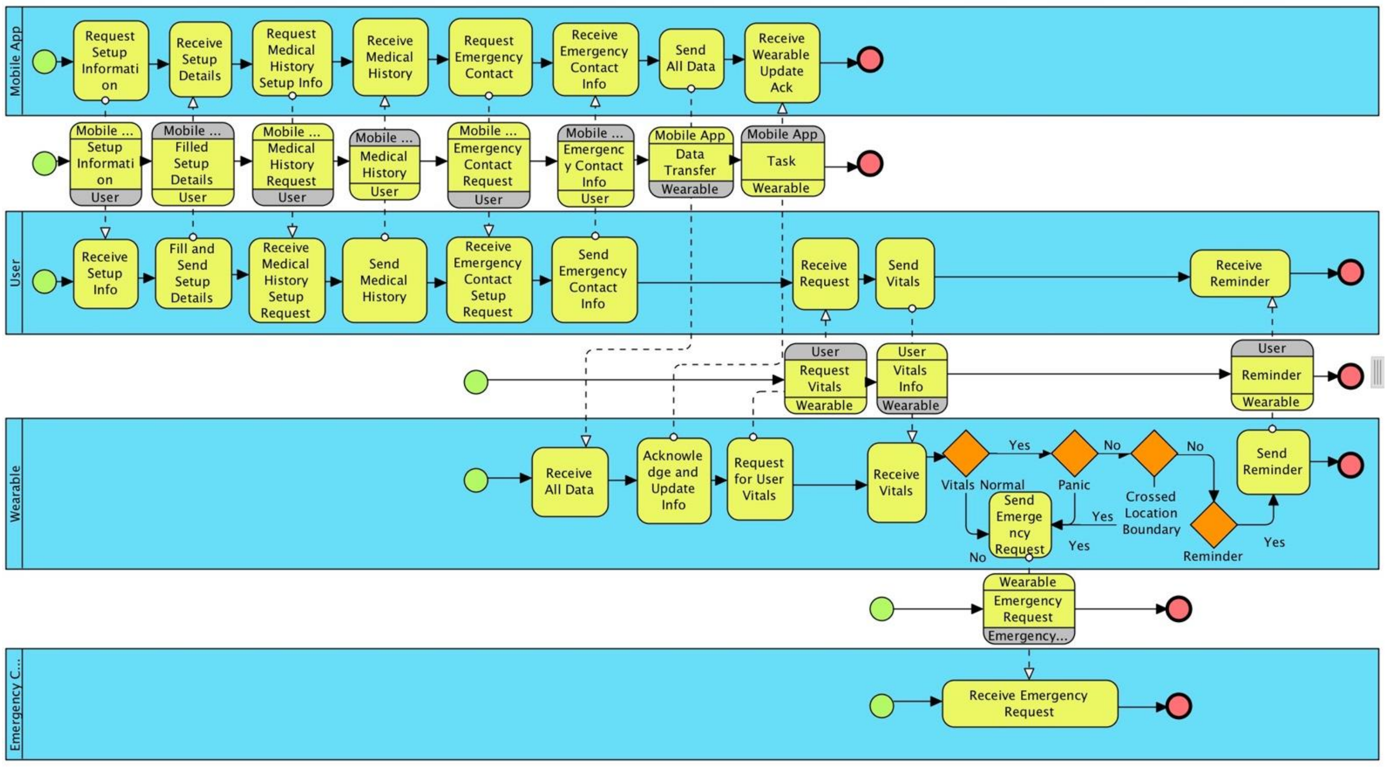
PROPERTY OF APURVA GODGHASE

Functional Specification

- User buys the smartwatch and downloads the Mobile Application. User will register on mobile application by specifying basic information, emergency contact details, threshold distance, and medical history
- The system will allow users to setup medicine reminders for multiple times in a day and for specific number of days. User can also specify repeat frequency for all reminders
- User can set reminders for Doctor's appointments
- Smartwatch will continuously track user's vitals – heartbeat, blood pressure, calories etc. And smartwatch will send vitals data to the system.
- The system will fetch generic vitals data from external systems. Proposed system will then compare the user's vitals data with generic vitals data.
- The proposed System has provision for Heart attack or abnormal cardiac rhythm prediction through recorded data. System will get generic data from heart attacks and compare it with user's heart rate. It will automatically alert emergency contacts if there's a high risk of heart attack
- Mobile Application and Smartwatch has panic button. In case of emergency, user can press this panic button and a panic alert will be sent to emergency contacts
- The proposed system provides functionality of location tracking. This functionality is useful for users having Alzheimer's disease. Their family members can setup threshold distance (for example, 500 meters) at the time of registration. If a user goes beyond that distance, a location cross alert will be sent to emergency contacts

PROPERTY OF APURVA GODGHASE

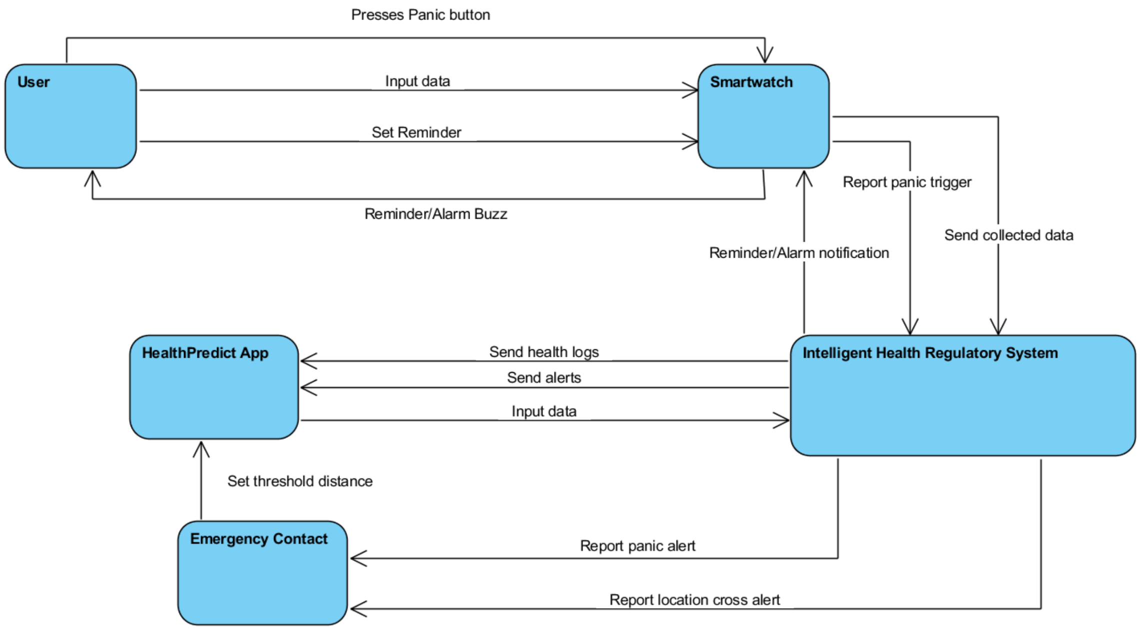
Business Process Model and Notation



Mobile App
Request Setup Information
Receive Setup Details
Request Medical History Setup Info
Receive Medical History
Request Emergency Contact
Receive Emergency Contact Info
Send All Data
Receive Wearable Update Ack
Mobile ... Setup Information User
Mobile ... Filled Setup Details User
Mobile ... Medical History Request User
Mobile ... Medical History User
Mobile ... Emergency Contact Request User
Mobile ... Emergency Contact Info User
Mobile App Data Transfer Wearable
Mobile App Task Wearable
User
Receive Setup Info
Fill and Send Setup Details
Receive Medical History Setup Request
Send Medical History
Receive Emergency Contact Setup Request
Send Emergency Contact Info
Receive Request
Send Vitals
Receive Reminder
User Request Vitals Wearable
User Vitals Info Wearable
User Reminder Wearable
Wearable
Receive All Data
Acknowledge and Update Info
Request for User Vitals
Receive Vitals
Vitals Normal
Yes
No
Panic
No
Crossed Location Boundary
No
Yes
Send Emergency Request
Yes
Reminder
Yes
Send Reminder
Wearable Emergency Request Emergency...
Emergency C...
Receive Emergency Request

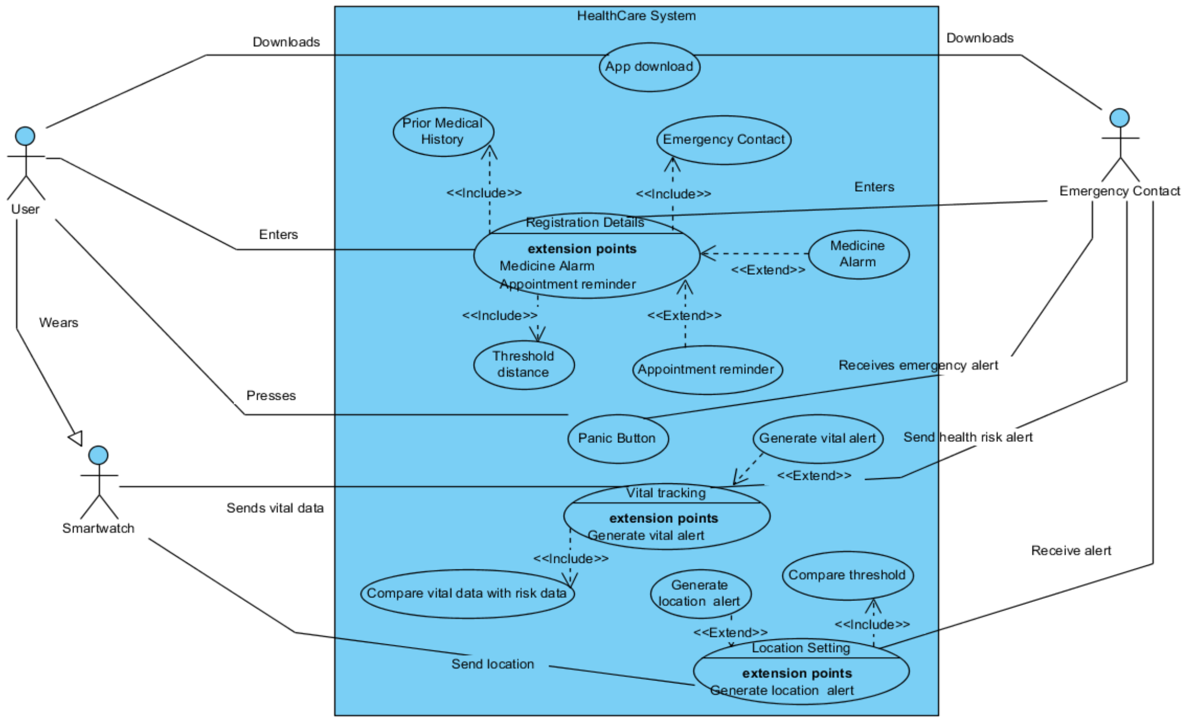
PROPERTY OF A

System Context Diagram



PROPERTY OF APURVA

Use Case Diagram



PROPERTY OF APU

Use Case Description

1. Use Case Description 1:

Use Case Name: Application download
Primary Actor: User, Emergency Contact
Stakeholders: Smartwatch
Description: User downloads and installs the app
Trigger: Customer(User)
Precondition: User has downloaded the app on the smartwatch
Normal flow of events:
1. User visits app store
2. User searches the app
3. User downloads the app
4. User Logs into the account

2. Use Case Description 2:

Use Case Name: Register Account
Primary Actor: User, Emergency Contact
Stakeholders: HealthReflex App
Description: User creates a new account
Trigger: Customer(User) opens the app
Precondition: User has downloaded the app
Normal flow of events:
1. User opens the HealthReflex App.
2. User enters the Personal details into the App.
3. User enters Prior medical history details into the App
4. User enters Emergency contact details into the App
5. User sets the safe radius
6. App authenticates the details entered.
Exceptional Flow:
2A. If Personal Details, Prior medical history details, Emergency Contact details are empty or contains invalid values, then display “Registration Unsuccessful! Please enter valid details”.

3. Use Case Description 3:

Use Case Name: Set Remainder
Primary Actor: User
Stakeholders: Smartwatch
Description: User sets remainder about the appointment
Trigger: Appointment reminder option is selected/ voice instructions are received
Precondition: User is in proximity to the watch and has app installed on the smartwatch

PROPERTY OF APURVA GODGHASE

Normal flow of events:
1. User communicates with the watch to set the alarm
2.User specifies the reminder specifics
3. Watch displays the 'Reminder set successfully' message
Exceptional Flow:
2A. If there's existing reminder at the same time, display 'Reminder already exists' If the alarm details are in the past then display 'Invalid time'

4. Use Case Description 4

Use Case Name: Track Vitals
Primary Actor: Smartwatch
Stakeholders: User
Description: Smartwatch continuously tracks the vitals of the user and sends to the system
Trigger: Person wears the smartwatch
Precondition: Person has worn the smartwatch
Normal flow of events:
1. User wears the smartwatch
2. App continuously tracks vitals and sends to the system

5. Use Case Description 5

Use Case Name: Panic alert
Primary Actor: User
Stakeholders: Emergency Contact, Smartwatch
Description: User sends panic alert
Trigger: User presses the panic button
Precondition: User has emergency health issue and needs help
Normal flow of events:
1. User presses the panic button
2. Emergency contact receives the panic alert

6. Use Case Description 6

Use Case Name: Location cross alert
Primary Actor: User
Stakeholders: Emergency Contact, Smartwatch
Description: User crosses the safe zone
Trigger: User has crossed the threshold distance
Precondition: User location is outside the safe location distance
Normal flow of events:
1. User moves to a location that is outside the previously specified safe radius

PROPERTY OF APURVA GODGHASE

2. Emergency contact receives the location crossing alert

7. Use Case Description 7

Use Case Name: Compare threshold
Primary Actor: Smartwatch
Stakeholders: User
Description: System compares the current location with the threshold location
Trigger: Current Location is received
Precondition: GPS is active on the watch
Normal flow of events:
1. Current location of the system is received
2. Current location parameters are compared with the safe radius
Exceptional Flow:
If GPS is not active, send a message for seeking permission to turn on the GPS

8. Use Case Description 8

Use Case Name: Compare vitals
Primary Actor: Smartwatch
Description: The system compares the vitals
Trigger: User Vital details are received
Precondition: Smartwatch is active
Normal flow of events:
1. Track Vitals are received from the smartwatch
2. Vital details are compared with the safe threshold range

PROPERTY OF APURVA GODGHASE

Data Dictionary

Use Case 1 : Application Download

Login = Username + Password
Username = data element
Password = data element

Use Case 2 : Register Account

Personal Details = FirstName + LastName + Age + Sex + UserContactNumber + Useremail
FirstName= data element
LastName = data element
Age = data element
Sex = data element
UserContactNumber = data element
Useremail = data element
Prior medical history details = MedicalHistory
MedicalHistory = data element
Emergency Contact Details = {ContactName + ContactNumber + ContactAddress}2
ContactName = data element
ContactNumber = data element
ContactAddress = data element
Safe Radius = DistanceThreshold
DistanceThreshold = data element

Use Case 3 : Set Reminder

Reminder Specifics = Day + FromDate + ToDate + FromTime + ToTime + ReminderDetails + RepeatFrequency
Day = data element
FromDate = data element
ToDate = data element
FromTime = data element
ToTime = data element
ReminderDetails = data element
RepeatFrequency = data element

Use Case 4 : Track Vitals

TrackVitals = Heartbeat + Calories + VitalDate + VitalTime
HeartBeat = data element
Calories = data element
VitalDate = data element
VitalTime = data element

PROPERTY OF APURVA GODGHASE

Use Case 5 : Panic Alert

Emergency Contact Details = ContactName + ContactNumber + ContactAddress
ContactName = data element
ContactNumber = data element
ContactAddress = data element

Use Case 6 : Location Cross Alert

Safe Radius = DistanceThreshold
DistanceThreshold = data element

Use Case 7 : Compare Threshold

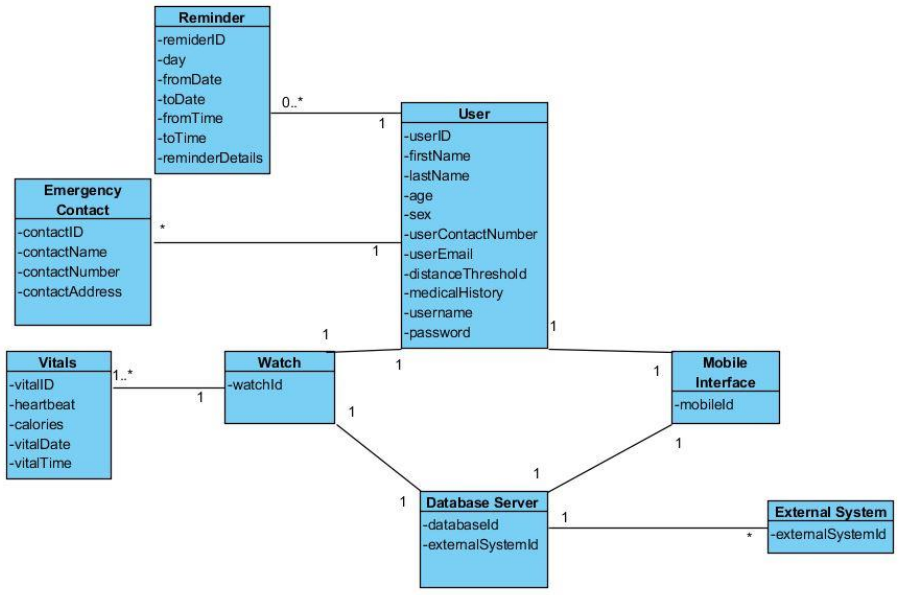
Safe Radius = DistanceThreshold
DistanceThreshold = data element

Use Case 8 : Compare Vitals

TrackVitals = Heartbeat + Calories + VitalDate + VitalTime
HeartBeat = data element
Calories = data element
VitalDate = data element
VitalTime = data element

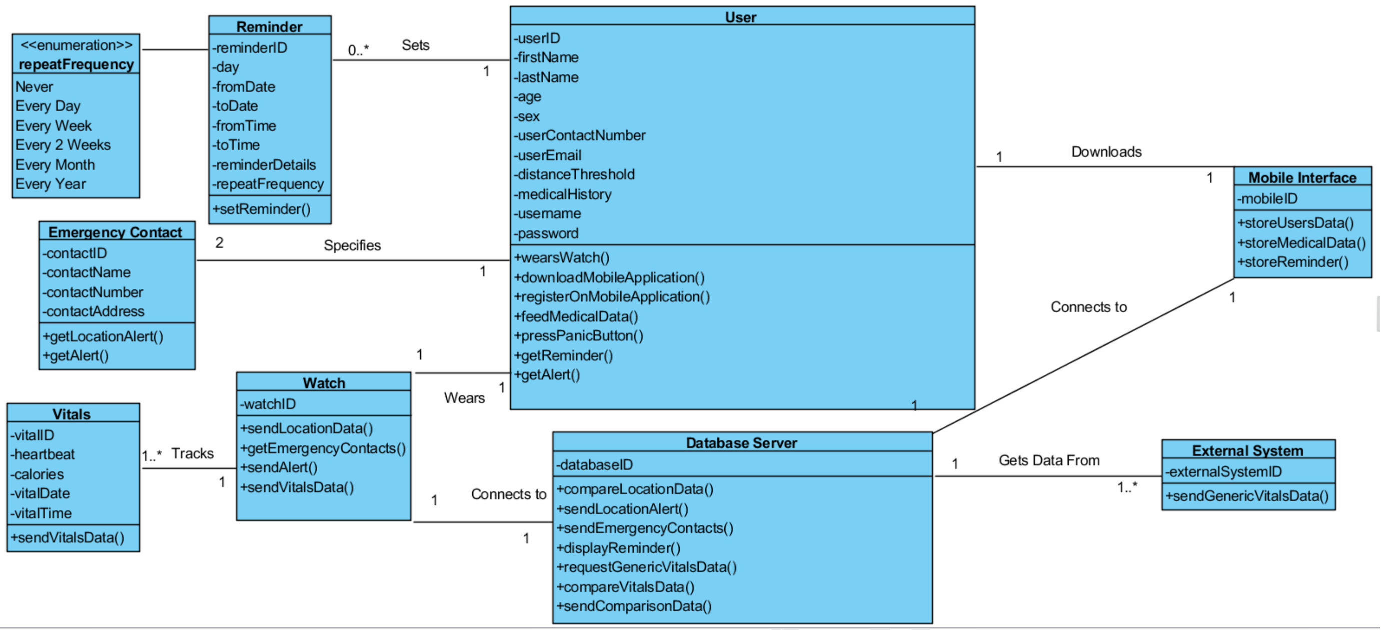
PROPERTY OF APURVA GODGHASE

Data Model Diagram



PROPERTY OF APURVA

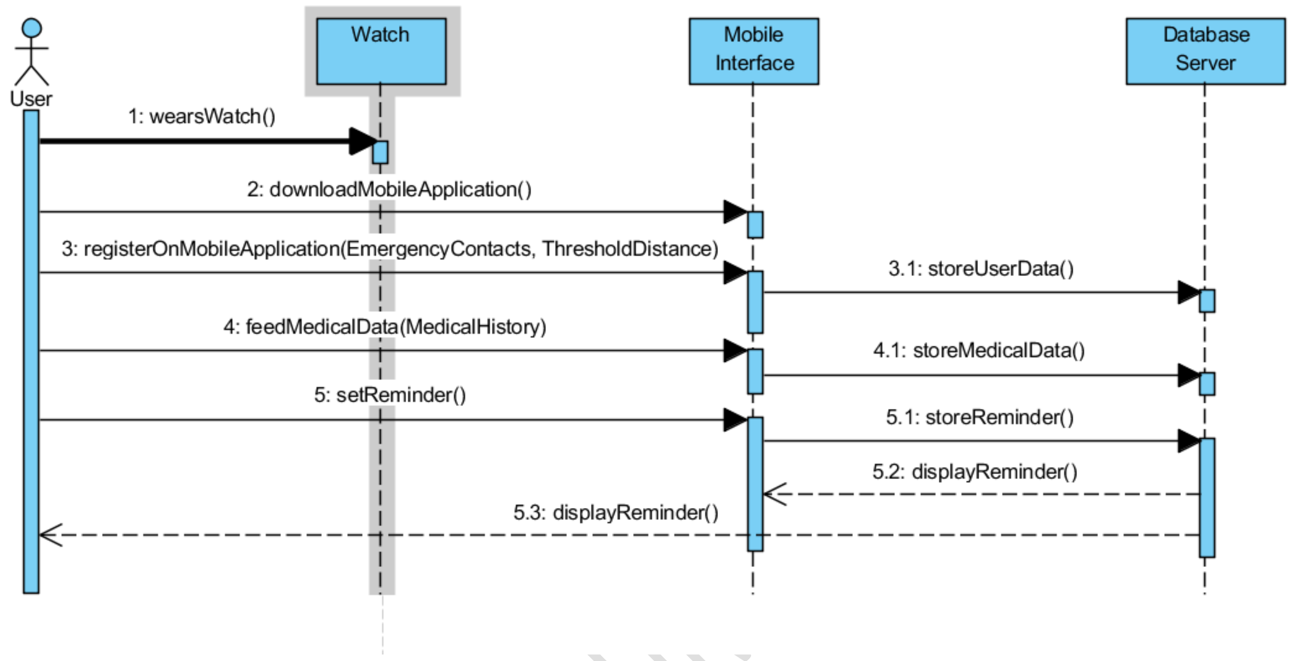
Class Diagram with Methods



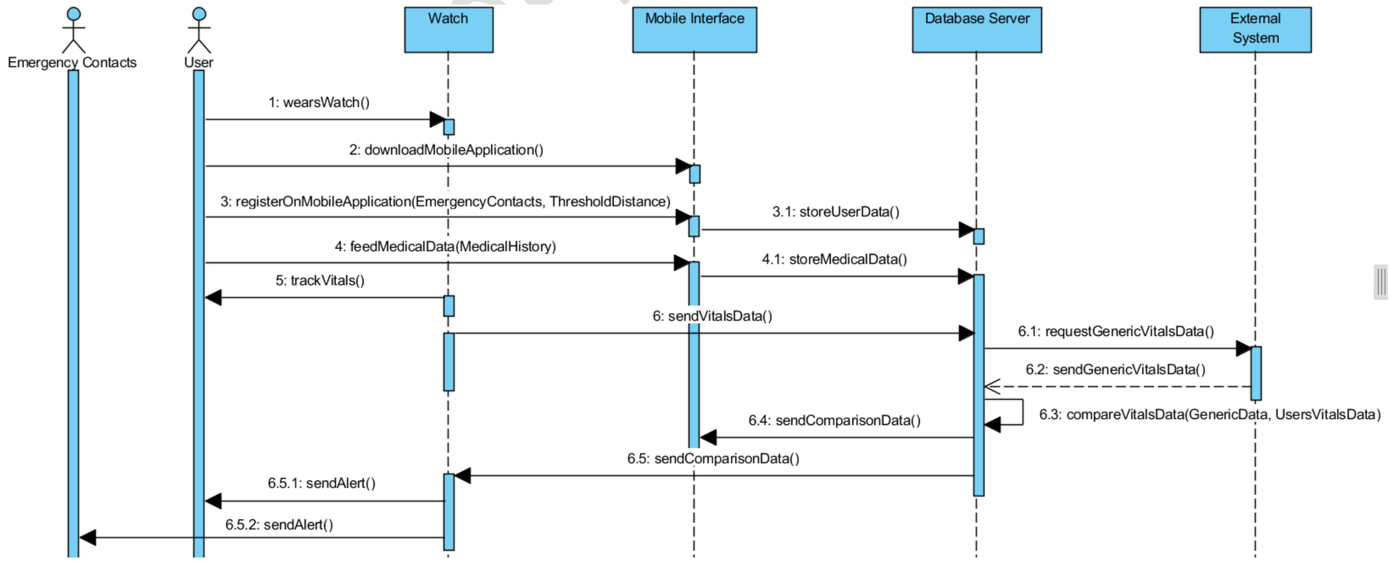
PROPERTY OF APURVA

Sequence Diagram

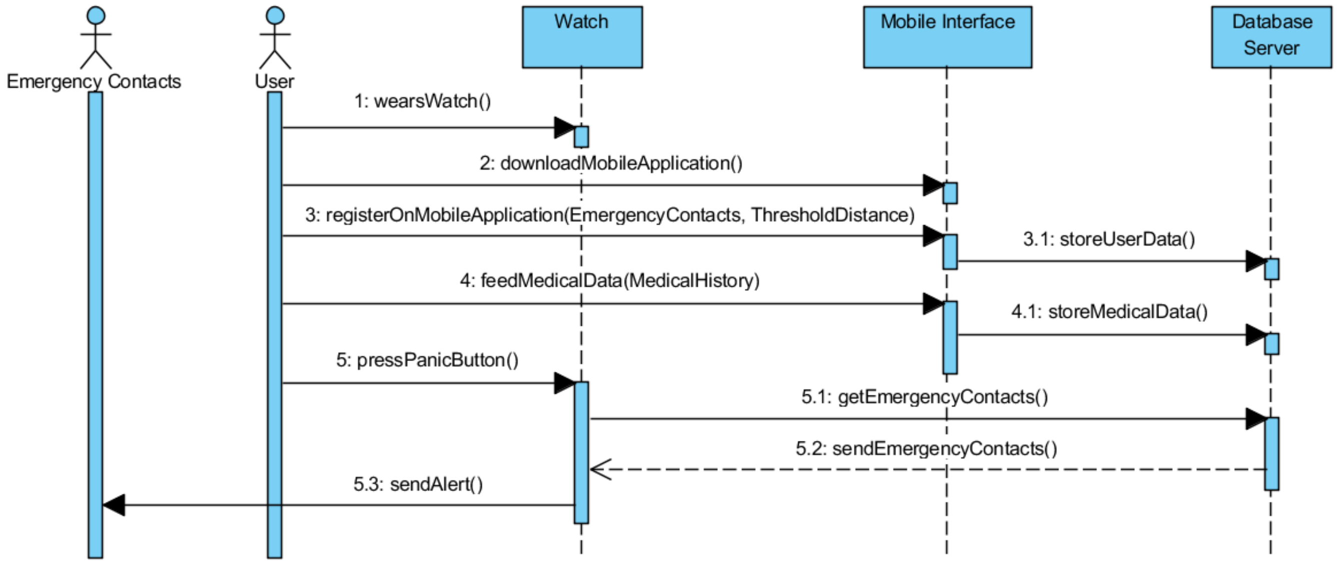
1. Set Reminders



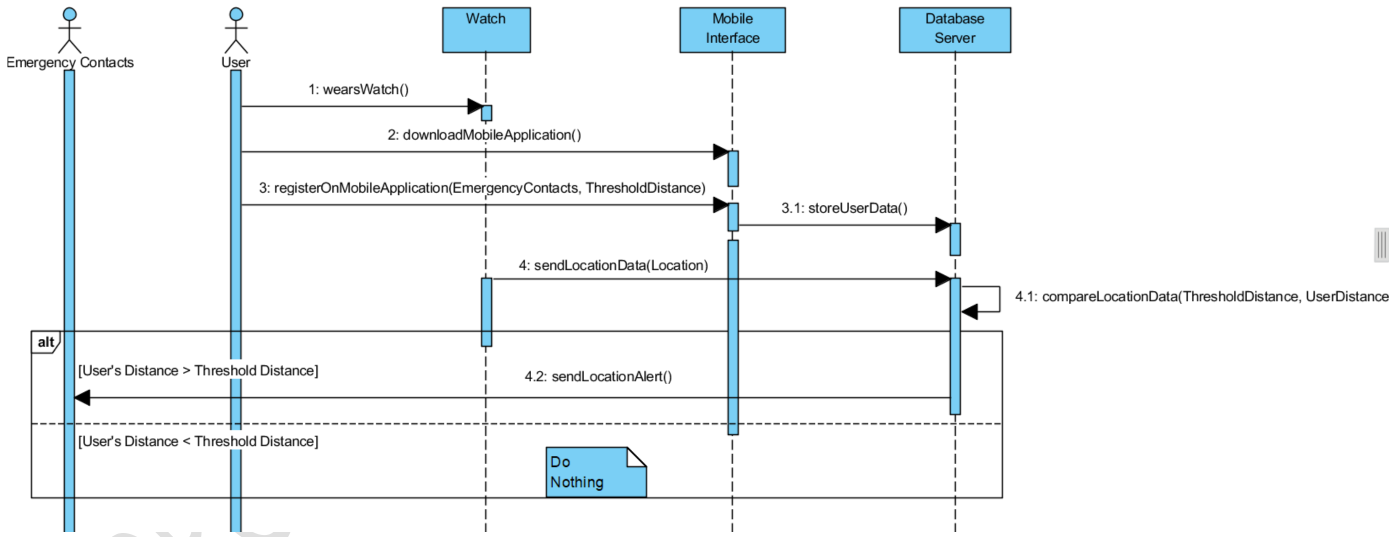
2. Track Vitals


Emergency Contacts
User
Watch
Mobile Interface
Database Server
External System
1: wearsWatch()
2: downloadMobileApplication()
3: registerOnMobileApplication(EmergencyContacts, ThresholdDistance)
3.1: storeUserData()
4: feedMedicalData(MedicalHistory)
4.1: storeMedicalData()
5: trackVitals()
6: sendVitalsData()
6.1: requestGenericVitalsData()
6.2: sendGenericVitalsData()
6.3: compareVitalsData(GenericData, UsersVitalsData)
6.4: sendComparisonData()
6.5: sendComparisonData()
6.5.1: sendAlert()
6.5.2: sendAlert()

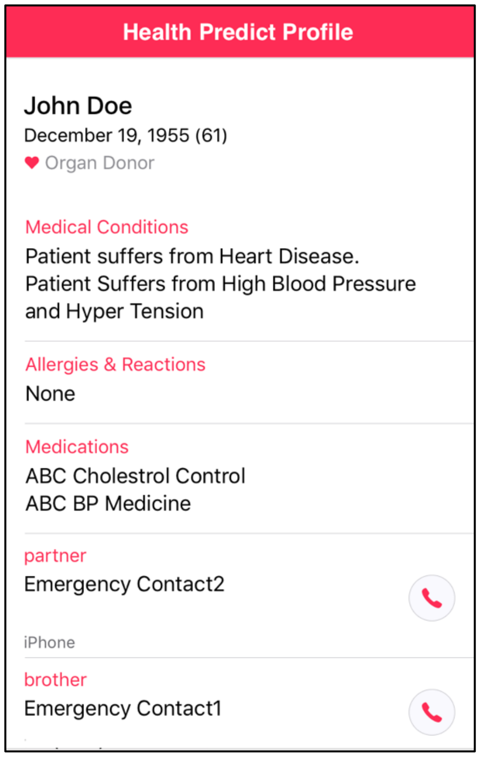
3. Panic Button


Emergency Contacts
User
Watch
Mobile Interface
Database Server
1: wearsWatch()
2: downloadMobileApplication()
3: registerOnMobileApplication(EmergencyContacts, ThresholdDistance)
3.1: storeUserData()
4: feedMedicalData(MedicalHistory)
4.1: storeMedicalData()
5: pressPanicButton()
5.1: getEmergencyContacts()
5.2: sendEmergencyContacts()
5.3: sendAlert()

4. Location Tracking


Emergency Contacts
User
Watch
Mobile Interface
Database Server
1: wearsWatch()
2: downloadMobileApplication()
3: registerOnMobileApplication(EmergencyContacts, ThresholdDistance)
3.1: storeUserData()
4: sendLocationData(Location)
4.1: compareLocationData(ThresholdDistance, UserDistance
alt
[User's Distance > Threshold Distance]
4.2: sendLocationAlert()
[User's Distance < Threshold Distance]
Do Nothing

Interface Design



Health Predict Profile

John Doe
December 19, 1955 (61)
♥ Organ Donor

Medical Conditions
Patient suffers from Heart Disease.
Patient Suffers from High Blood Pressure and Hyper Tension

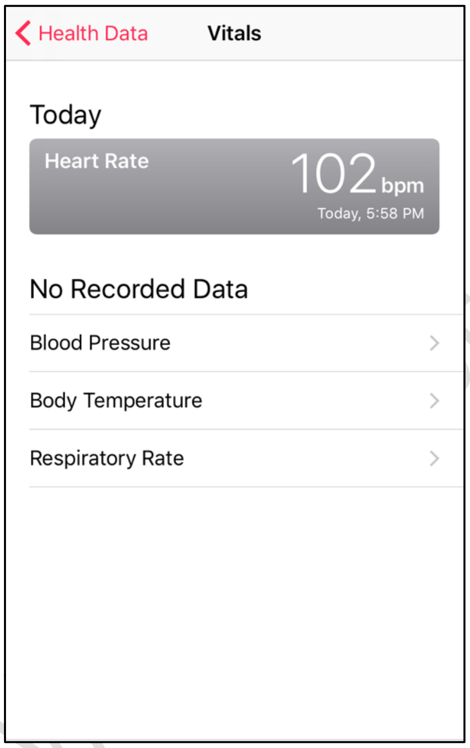
Allergies & Reactions
None

Medications
ABC Cholestrol Control
ABC BP Medicine

partner
Emergency Contact2
iPhone

brother
Emergency Contact1

Image 1 - Profile Setup



Health Data Vitals

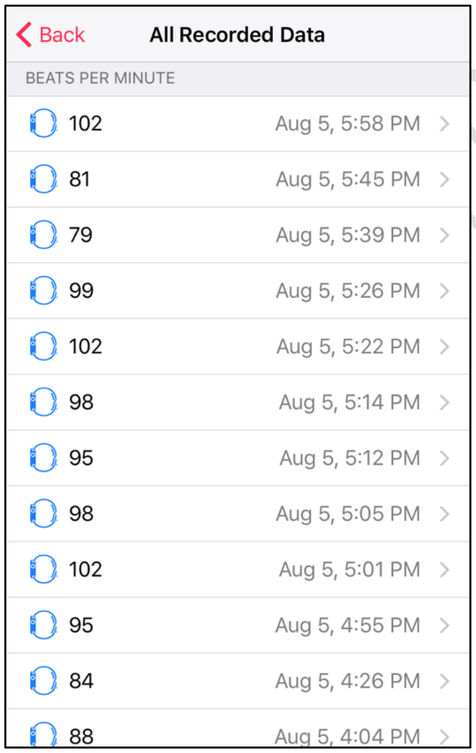
Today

Heart Rate 102 bpm
Today, 5:58 PM

No Recorded Data

Blood Pressure
Body Temperature
Respiratory Rate

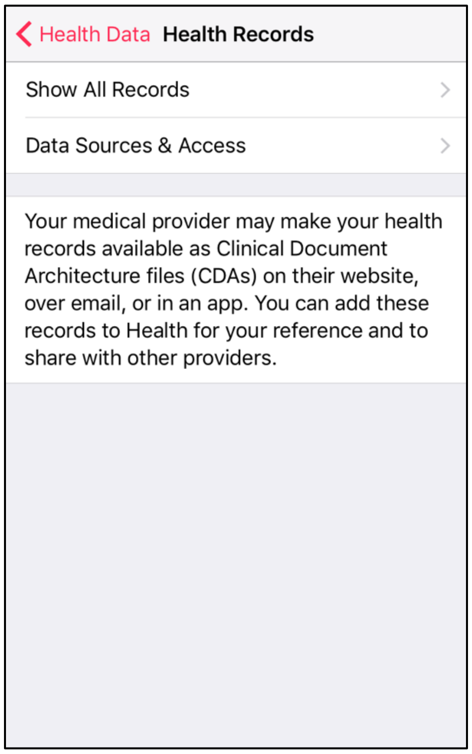
Image 2 - Vitals Recording



Back All Recorded Data

BEATS PER MINUTE	
102	Aug 5, 5:58 PM
81	Aug 5, 5:45 PM
79	Aug 5, 5:39 PM
99	Aug 5, 5:26 PM
102	Aug 5, 5:22 PM
98	Aug 5, 5:14 PM
95	Aug 5, 5:12 PM
98	Aug 5, 5:05 PM
102	Aug 5, 5:01 PM
95	Aug 5, 4:55 PM
84	Aug 5, 4:26 PM
88	Aug 5, 4:04 PM

Image 3 - Heartbeat Monitoring

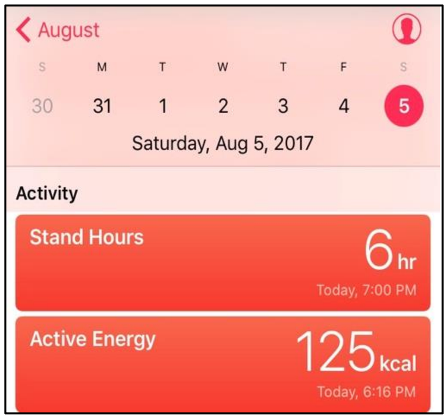


Health Data Health Records

Show All Records
Data Sources & Access

Your medical provider may make your health records available as Clinical Document Architecture files (CDAs) on their website, over email, or in an app. You can add these records to Health for your reference and to share with other providers.

Image 4 - Health Records of Patient



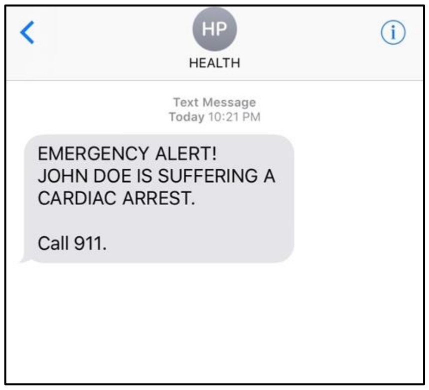
Saturday, Aug 5, 2017

Activity

Stand Hours 6 hr
Today, 7:00 PM

Active Energy 125 kcal
Today, 6:16 PM

Image 5 - Vitals: Calorie Monitoring

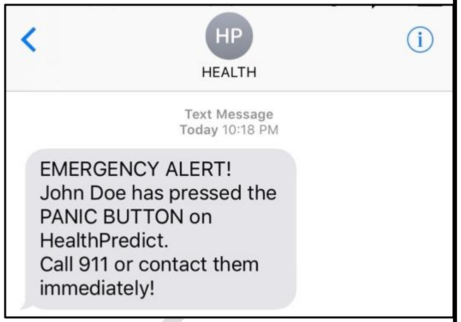


Text Message
Today 10:21 PM

EMERGENCY ALERT!
JOHN DOE IS SUFFERING A CARDIAC ARREST.

Call 911.

Image 6 - Emergency Alert 1



Text Message
Today 10:18 PM

EMERGENCY ALERT!
John Doe has pressed the PANIC BUTTON on HealthPredict.
Call 911 or contact them immediately!

Image 7 - Emergency Alert 2

PROPERTY OF APURVA GODGHASE

Database Design

1. **Watch** (WatchID)
   - WatchID should be unique and not NULL
2. **User** (UserID, Username, Password , FirstName, LastName, Age, Sex, UserContactNumber, Useremail, ThresholdDistance, MedicalHistory, *EmergencyID*, *WatchID*, *LoginID, VitalID, GenericID, LocationID*)
   - UserID should be unique and not NULL
   - LoginID should be non-null and should exist in the UserLogin table.
   - VitalID should be non-null and should exist in the Vital table.
   - GenericID should be non-null and should exist in the GenericData table.
   - LocationID should be non-null and should exist in the Location table.
   - WatchID should be non-null and should exist in the Location table.
3. **EmergencyContact** (EmergencyID, ContactName, ContactNumber, ContactAddress, *UserID*)
   - EmergencyID should be unique and not NULL
   - UserID should be non-null and should exist in the Location table.
4. **Reminder** (ReminderID, Day, FromDate, ToDate, FromTime, ToTime, ReminderDetails, RepeatFrequency, *UserID*)
   - ReminderID should be unique and not NULL
   - UserID should be non-null and should exist in the Location table.
5. **Vitals** (VitalID, Heartbeat, Calories, VitalDate, VitalTime, *WatchID*)
   - VitalID should be unique and not NULL
   - WatchID should be non-null and should exist in the Location table.

PROPERTY OF APURVA GODGHASE

Software Design

1. **Signature**

Method Name: Set Reminder()	Class Name: Reminder	ID: reminderID
Clients (Customers): User, MobileApplication		
Associated Use Cases: Set Appointment Reminder, Set Medicine Reminder Alarm		
Description of Responsibilities: 1. User sets reminder for doctor's appointment 2. User sets medicine reminders for multiple times in a day and specific number of days		
Arguments Received: ReminderDetails, day, fromDate, toDate, fromTime, toTime, repeatFrequency		
Type of Value Returned: Alert on Mobile Application, Alert on Smartwatch		
Pre-Conditions: 1. User is in proximity to the watch and has application installed on the smartwatch 2. User Clicks on Set Reminder button in the mobile application 3. Reminder details should not be NULL 4. fromDate, toDate, fromTime, toTime should be valid		
Post-Conditions: Watch displays the 'Reminder set successfully' message		

Logic:

```
IF User clicks on Set Reminder
THEN FETCH Time from System Date
ACCEPT ReminderDetails
ACCEPT day
ACCEPT fromDate
ACCEPT toDate
ACCEPT fromTime
ACCEPT toTime
ACCEPT repeatFrequency
END IF
IF ((fromDate < Today()) && (todate < Today()))
    DISPLAY message 'Invalid Date'
ELSE IF CheckReminder(fromDate, toDate, fromTime, toTime) == TRUE
    DISPLAY message 'Alarm already exists'
ELSE PROCESS Set new Reminder END IF
```

PROPERTY OF APURVA GODGHASE

2. Signature

Method Name: Send Vitals Data()	Class Name: Vitals	ID: vitalID
Clients (Customers): User, Smartwatch		
Associated Use Cases: Track Vitals		
Description of Responsibilities: Smartwatch continuously tracks the vitals of the user and sends to the system		
Arguments Received: heartbeat, calories, vitalDate, vitalTime		
Type of Value Returned: None		
Pre-Conditions: Person has worn the smartwatch		
Post-Conditions: Sending the user's vitals to the system		

Logic:

```
DO (for every 10 seconds)
FETCH Heartbeat
SET Heartbeat = heartbeat INTO Vitals Table
FETCH Calories
SET Calories = calories INTO Vitals Table
FETCH Date FROM System Date
SET vitalDate = Date INTO Vitals Table
FETCH Time FROM System Date
SET vitalTime = Time INTO Vitals Table
WHILE User is wearing Smartwatch
END DO
```

PROPERTY OF APURVA GODGHASE

3. Signature

Method Name: Compare Vitals Data()	Class Name: Database Server	ID: database_ID
Clients (Customers): Database Server, External System, User, Emergency Contacts		
Associated Use Cases: Compare Vitals		
Description of Responsibilities: Compares user’s vitals with generic vital data and if there’s any similarity, it will alert the user.		
Arguments Received: User’s Vitals data, Generic Vitals Data		
Type of Value Returned: Text Message Alert, Alert on Mobile Application, Alert on Smartwatch		
Pre-Conditions: System has received user’s vitals data		
Post-Conditions: Sending the alert to emergency contacts via text message. Sending the alert to User on Smartwatch		

Logic:
```
DO (for every 10 seconds)
FETCH Heartbeat FROM Vitals Table
FETCH Generic Vitals Data FROM External System
IF Heartbeat != Generic Vitals Data.Resting Heart Beat
DISPLAY alert message TO User ON Smartwatch
     DISPLAY alert message TO Emergency Contacts
ELSE
DO NOTHING
```

PROPERTY OF APURVA GODGHASE

4. Signature

Method Name: Press Panic Button()	Class Name: User	ID: userID
Clients (Customers): User, Emergency Contacts		
Associated Use Cases: Panic Alert		
Description of Responsibilities: When user presses panic button, alert is sent to emergency contacts		
Arguments Received: None		
Type of Value Returned: Text Message alert		
Pre-Conditions: User presses the Panic Button		
Post-Conditions: Panic alert is sent to emergency contacts via text message		

Logic:

```
IF User presses Panic Button
THEN
FETCH Emergency Contact Name FROM Emergency Contact Table
FETCH Emergency Contact Number FROM Emergency Contact Table
DISPLAY panic alert message TO Emergency Contacts ON Emergency
Contact Number
```

PROPERTY OF APURVA GODGHASE

5. Signature

Method Name: Send Location Alert()	Class Name: Database Server	ID: databaseID
Clients (Customers): User, Emergency Contacts		
Associated Use Cases: Compare Threshold, Location Cross Alert		
Description of Responsibilities: When user goes beyond the threshold distance, alert is sent to emergency contacts		
Arguments Received: arrival timestamp, departure timestamp, section		
Type of Value Returned: time		
Pre-Conditions: User has crossed the threshold distance		
Post-Conditions: Location cross alert is sent to emergency contacts		

Logic:

```
IF User moves to a location that is outside the previously
specified threshold distance
THEN
FETCH Emergency Contact Name FROM Emergency Contact Table
FETCH Emergency Contact Number FROM Emergency Contact Table
DISPLAY location cross alert message TO Emergency Contacts ON
Emergency Contact Number
```

PROPERTY OF APURVA GODGHASE

Timeline

Execution Timeline:

Weekly Schedule		Task
5/28/17	6/3/17	Team introduction
6/4/17	6/10/17	Discussed Project Idea
6/11/17	6/13/17	Finalized topic
6/14/17	6/17/17	Discussed Context Diagram
6/18/17	6/24/17	Problem Documentation and Use Case
6/25/17	7/1/17	Use Case Descriptions and Sequence diagram
7/2/17	7/8/17	Data Dictionary notations and BPMN
7/9/17	7/15/17	Class Diagrams and Database Design
7/16/17	7/22/17	Report Documentation and Interface design
7/30/17	8/5/17	Finalized Report

Planned Timeline:

Weekly Schedule		Task
5/28/17	6/3/17	Introduction to team
6/4/17	6/10/17	Discuss Project Idea
6/11/17	6/13/17	Finalize topic and Context Diagram
6/14/17	6/17/17	Problem Documentation and Use Case
6/18/17	6/24/17	Use Case description for Feature 1
6/25/17	7/1/17	Use Case description for Feature 2
7/2/17	7/8/17	Data Dictionary notations and BPMN
7/9/17	7/15/17	Class Diagrams and Database Design
7/16/17	7/22/17	Report Documentation and Interface design
7/30/17	8/5/17	Finalize Report

PROPERTY OF APURVA GODGHASE

Minutes of Meeting

Agenda	Date	Time	Next Meeting
Introduction to team	05/30/17	7pm-7:30pm	06/05/17
Project Topic Discussion. Identifying practical feasibility of the shortlisted topics	06/05/17	7pm-8pm	06/24/17
Came up with first draft for the project topic	06/24/17	7pm-9pm	06/30/17
Came up with second draft for the project topic	06/30/17	7pm-9pm	07/03/17
Problem Documentation and draw the Context Diagram	07/05/17	8pm-9:30pm	07/08/17
Finalize Use Case Description for Feature 1	07/08/17	4pm-7pm	07/10/17
Data Dictionary notations and BPMN notations	07/11/17	8pm-9pm	07/15/17
Class Diagrams and Database Design	07/15/17	6pm-10pm	07/20/17
Finalized Sequence Diagram	07/20/17	6pm-10pm	07/25/17
Finalized the Project Summary and Problem Statement	07/26/17	8pm-9pm	07/29/17
Report Documentation and Interface design	07/29/17	4pm-7pm	

PROPERTY OF APURVA GODGHASE

References

- ER Diagram in Visio: https://www.youtube.com/watch?v=knvE3L57qrI
- Class Diagram in Visio: https://www.youtube.com/watch?v=3fjjgBRtXPI
- "Object-Oriented Systems Analysis and Design" by Jeff Hoffer, Joey George, and Joe Valacich, Pearson Prentice-Hall, Second Edition, 2006
- FTC Mobile Health Apps Interactive Tool "https://www.ftc.gov/tips-advice/business-center/guidance/mobile-health-apps-interactive-tool"
- Mobile Health Apps – Implications on Healthcare and Opportunities for Regulatory Affairs "https://teamlead.duke-nus.edu.sg/vapfiles_ocs/core/core_mobile_health_app_implication_on_healthcare_opportunities_for_regulatory_affairs/index5.html"
- Interface Design built over **Apple Health app** for Ios

PROPERTY OF APURVA GODGHASE

PROPERTY OF APURVA GODGHASE