# Design Document – E-Health Workshop - “Hydrapp”

Name of the project: Hydrapp

Name of the team members + studentID + email

#1: Noam Weinman 302819750 [noamweinman@gmail.com](mailto:noamweinman@gmail.com)

#2: Ben Sterenson 304745375 [bensterenson@gmail.com](mailto:bensterenson@gmail.com)

# Table of Contents

[Table of Contents 1](#_Toc230922059)

[Introduction 2](#_Toc230922060)

[1.1 Overview 2](#_Toc230922061)

[1.2 Design Goals and Non-Goals 2](#_Toc230922062)

[1.2.1 Goals 2](#_Toc230922063)

[1.2.2 Non-Goals 2](#_Toc230922064)

[1.3 Dependencies 2](#_Toc230922065)

[1.3.1 Feature Dependencies 2](#_Toc230922066)

[1.3.2 Features Dependent on this Feature 2](#_Toc230922067)

[1.4 Audience 2](#_Toc230922068)

[1.5 Assumptions and Design Constraints List 2](#_Toc230922069)

[1.6 Issues List 2](#_Toc230922070)

[1.7 To-do List 3](#_Toc230922071)

[2 Logical Architecture 4](#_Toc230922072)

[2.1 Application Context 4](#_Toc230922073)

[3 Design 5](#_Toc230922074)

[3.1 Classes 5](#_Toc230922075)

[3.1.1 Class Diagram 5](#_Toc230922076)

[3.1.2 Class details 5](#_Toc230922077)

[3.2 Flows 6](#_Toc230922078)

[3.3 Synchronization and Protection Mechanisms 6](#_Toc230922079)

[3.4 Backward Compatibility Support 6](#_Toc230922080)

[3.5 Setup 6](#_Toc230922081)

[4 Physical Architecture 6](#_Toc230922082)

[5 References 6](#_Toc230922083)

[6 Revision History 6](#_Toc230922084)

# Introduction

## Overview

Dehydration is a well-known danger, defined as a deficit of total body water. While mild dehydration can cause fatigue and dizziness, high levels of fluid loss will cause physical and mental deterioration and in extreme conditions even death. Army commanders, trip guides or lone travelers find it hard to keep track of the fluid level of their subordinates.

Hydrapp is the anti-dehydration application that specializes in monitoring fluid loss of any person as a result of physical effort or outdoor activity. Hydrapp’s built-in managing groups feature will allow trip guides, army commanders or anyone who want to monitor others live physical condition, including dehydration levels. The App works with Microsoft Band to push live notification to the group’s admin, notifying the user on any abnormal condition their subordinates are facing.

## Design Goals and Non-Goals

### Goals

**Main Goal**

Hydrapp’s main goal is to give the user a wide scale overview of his group’s live physical condition. We will use the data, gathered from Microsoft Band of each participant, and show it to the admin user’s main view. The innovative idea behind Hydrapp is using the data provided from the band to give estimation of percentage of fluid loss. We want the application to be easy to use and straightforward to provide the admin user fast access to any important measures of his subordinates, as dehydration can be dangerous if not treated immediately.

**Secondary Goal**

Another important goal of Hydrapp is to provide personal recommendations to the admin user. The admin will be given notifications based on his history activities like: patterns of dehydration among his group participants, intense level of the workout compared to historic activities and so on.

### Non-Goals

Hydrapp will not aim to give exact physical data of the users, but a good assumption and close enough estimation. Using the recommendation system is not a substitute for drinking water.

## Dependencies

### Feature Dependencies

* Microsoft Band – a working band that the users wear on his wrist.
* Azure cloud services – an available access to azure cloud that stores all data.
* Mobile phone with Bluetooth – An available Bluetooth connection between mobile phone and band in order to retrieve body data extracted from band.
* Internet access – Hydrapp app depends on an available internet access in order to monitor and send data to cloud.
* User weight and height – in order to get best prediction Hydrapp needs current user weight & height.

## Assumptions and Design Constraints List

In order to get the best performance:

* Each user of the group must wear Microsoft Band on his wrist.
* Each user need to have the band connected to his phone.
* The phone should have an available internet connection.
* Hydrapp won’t support case sensitive input.

## Issues List

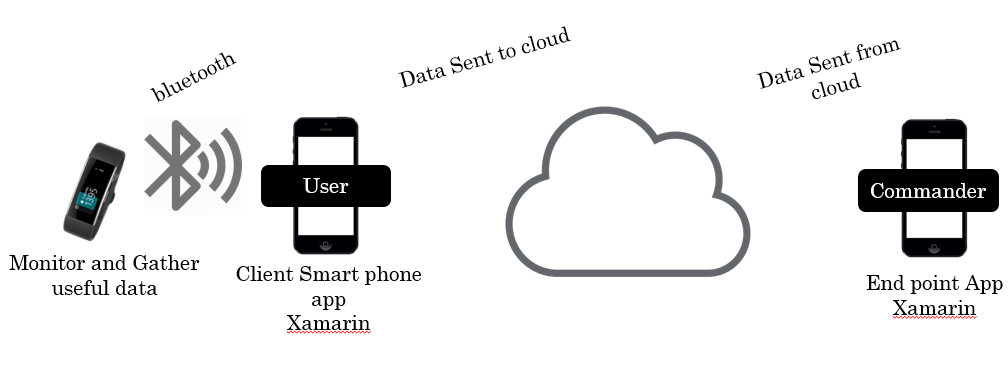
* Data accuracy - The accuracy provided from Microsoft Band can vary in different situation. Sometimes it is considered reliable, but when the band is not sitting perfect on the user’s hand, the data can change significantly. In order to deal with this issue, we will make sure all the users are wearing their bands tight and steady. Furthermore, we will try handle noise using software for example average on collected data.
* Formula use- We are using a formula which was given as a result of a research conducted in recent years regarding measuring dehydration level with electronic devices. We are relying on the formula, a risk which are taken into account. To minimize the risk we will try to adapt the formula to our tools and maybe change the weights of each parameter.
* Testing- In order to validate our calculated dehydration level, we will need a dehydrated person to test on him. We will try to get as close as we can to this situation and see if the application is giving a reasonable results.

## To-do List + Expected time-tables

|  |  |
| --- | --- |
| 15.1.17 | Detailed design document |
| 15.1.17 | Complete DB construction and integration with frontend |
| 15.1.17 | Service design and screens (UI) sketch |
| 22.1.17 | Service & Client Development |
| 22.1.17 | Demo presentation |
| 25.1.17 | Quality testing |
| 20.2.14 | Video + Website |
| 18.3.14 | Final testing and improvements |
| 30.3.14 | Deployment |

# Logical Architecture

## Application Context



# Design

## Classes

### Class Diagram

TODO

### Class details

**Front End Classes**

|  |  |
| --- | --- |
| **Class** | **Description** |
| LoginPage | Handles login process of the application. Authentication with Database stored on azure.  Logout action can be found at tool bar. |
| SignUpPage | Handles signup process for new User. |
| SettingPage | Allow User update information provided on signup. |
| GroupLoginPage | Allow user to:   * Create group - in that case the user becomes the Admin and the monitoring User. * Join existing group - - in that case the user is monitored by the Admin. |
| ManageGroupPage | **Admin** main page allow the admin to monitor all of the group users and receive notification in case of an abnormal fluid loss values. |
| MemberChartPage | Draws a graph of selected user fluid loss levels. |
| ChartsViewModel | View model that is responsible for binding data to charts. |
| ManageGroupPageViewModel | View model that is responsible for binding Users that joined admin group. |
| MainPage | **Monitored User** main page reads band values and sends extracted data to cloud for monitoring and analyzing. |
| MainPageViewModel | View model that is responsible for binding data extracted from band and sending to cloud. |

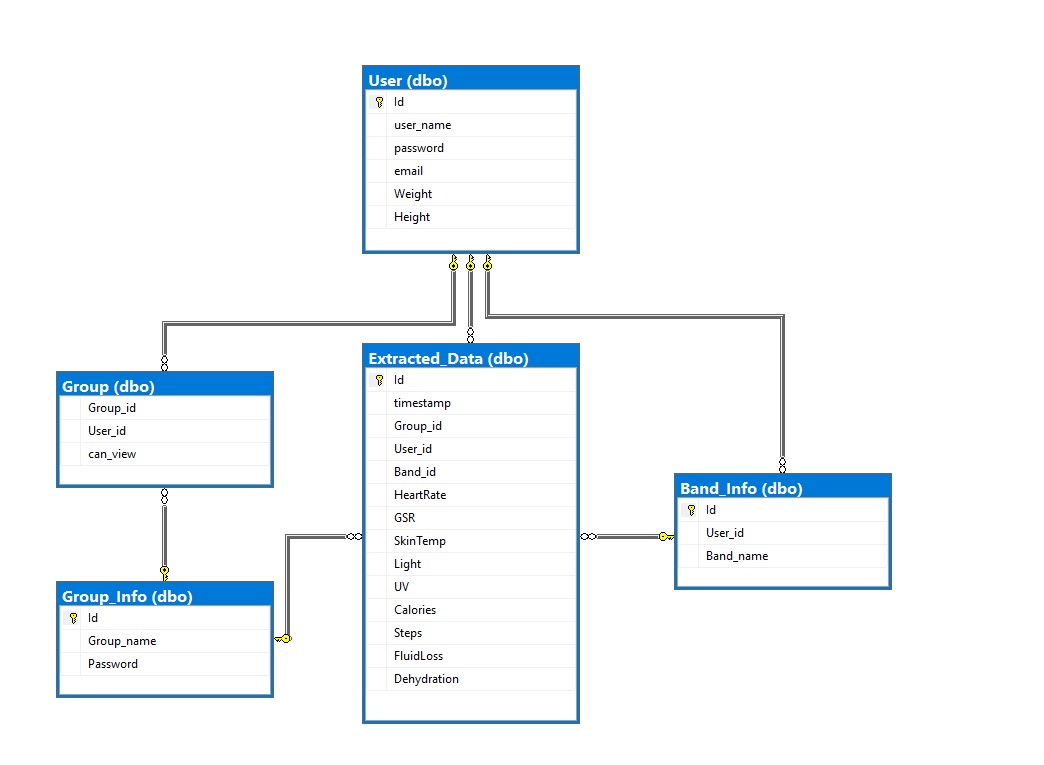
**Modules**

|  |  |
| --- | --- |
| **Class** | **Description** |
| User | Holds user information |
| Group | Holds group information |
| BandEntity | Holds band information extracted from cloud\band. |
| Participant | Holds participant information {user info, group info, band values read} |

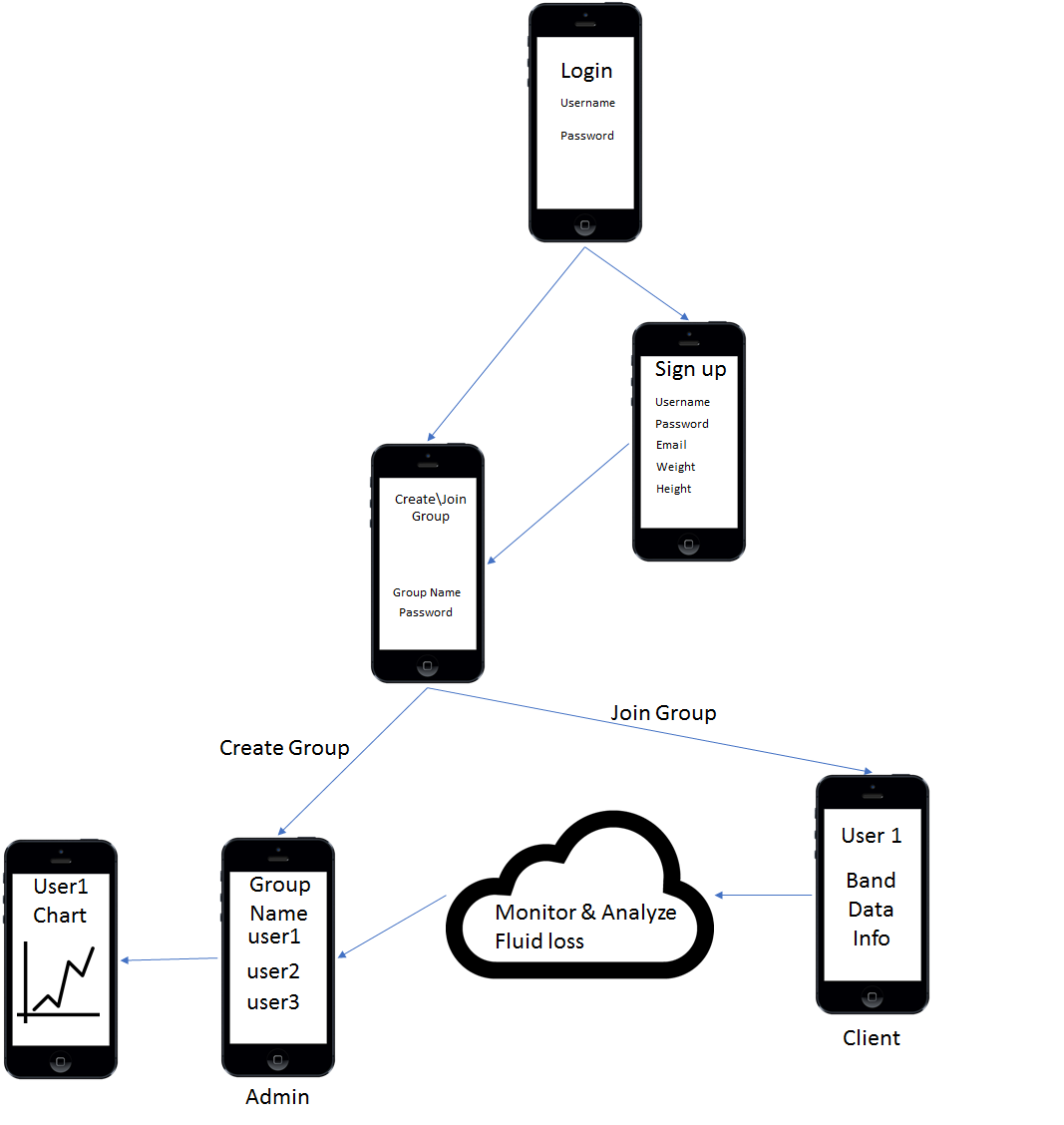
**Database service**

|  |  |
| --- | --- |
| TODO |  |
|  |  |
|  |  |
|  |  |

### Data Base design



## Flows



* Welcome screen - When app is starting
* Sign in/sign up - The user will sign in to the application with his credentials or create a new group as admin.
* Create group - The admin will put his user name, password and group name
* Sign up as User - The user will type his user name, password, group name, weight and height
* User home screen - The user will be able to start or stop sending data from his band.
* Admin home screen - The admin will view all his participants' data, including weather conditions. In this screen the admin can start and stop the activity. When starting the activity the data will be shown live, and when the stop button is pressed, the summary screen will be shown. Alerts will pop from this window.
* Settings Screen - The admin can add or remove alerts with parameters like: activity time, maximum value for specific measure and more. More settings are what to view in the home screen.
* Summary screen - summary of the activity
* For Admin:
  + First Use: Welcome screen -> Sign in/Sign up screen -> create group.
  + Normal Use: Welcome screen -> Sign in/Sign up screen -> Admin home screen -> Settings/Summary.
* For User:
  + First Use: Welcome screen -> Sign in/Sign up screen.
  + Normal Use: Welcome screen -> Sign in/Sign up screen -> User home screen.

## Synchronization and Protection Mechanisms

* Data is stored on cloud and are protected by could security mechanism.
* Input validation – verifying unnecessary data doesn’t exist in the system.
* Timeouts and exception handling

## Setup

* Server Application – Needs to be deployed on a cloud machine.
* Client Application – Needs to be installed on a smartphone.

# Physical Architecture

* Described in 2.1

# References – to external papers/packages

* Article  “Towards a Smart Non-Invasive Fluid Loss Measurement System” [Journal of Medical Systems](https://www.researchgate.net/journal/1573-689X_Journal_of_Medical_Systems) April 2015

<https://www.researchgate.net/publication/272519591_Towards_a_Smart_Non-Invasive_Fluid_Loss_Measurement_System>

# Revision History

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Description (include reviews with reviewer lists)** |
| 11.1.17 | Ben Sterenson  Noam Weinman | First Draft |
| 15.1.17 | Ben Sterenson | Added Flow picture, DB design |
|  |  |  |