# Introduction (Wei Lin & Brittany DePoi)

The Personal Health Assistant (PHA) Mobile Medical Management Application is a mobile application for both iOS and Android devices that allows the individual users to manage different aspects of their health, with a focus on managing medications. PHA (avoid using It at start of sentence) also gives the doctors and healthcare providers the ability to access patient these medications and other data which the patient authorizes to the provider. The goal of the project is to expand PHA to include data about disease management including diabetes, obesity, asthma, and chronic heart failure. This data can include diet, exercise, blood sugar levels, BP, pulse, etc. The data entry capability by patients will be augmented with the ability to deliver to healthcare providers give advance warning of potential threats in patient health by using advanced algorithms to detect potential health threats for the four diseases. The application is also capable of helping patients to manage their medication intake and alert the patient to take certain medication.

PHA will be extended with a disease management component developed using the Titanium platform. This allows the application to be easily translated between the iOS and Android operating systems, and even web access. The application also incorporates Microsoft’s HealthVault personal health record (PHR) to keep track of and interact with the personal health records. HealthVault allows the application to store sensitive health records in a secure database where the healthcare provider and the patient can only see what is allowed by the patient. Other open source EMR systems can also be incorporated such as Open mHealth, openEHR, and the SMART Framework. To illustrate, Figure 1 contains a conceptual diagram of the PHA components and their basic interactions.



Figure 1: System Overview of PHA and planned Extensions.

The Mobile Medical Management Application is an event driven application with three major components. The first is the collection of patient user interfaces on the patient’s personal mobile device. These interfaces capture patient information for a specific disease. Depending on the current health of the patient, the patient will have the option of downloading interfaces specifically for diabetes, heart disease, asthma or obesity. This information is then uploaded and saved to the Microsoft HealthVault database, where security is optimized using preexisting features. The healthcare provider can then access this information through the Internet, run optional diagnostics on the information and provide the patient with feedback. In addition, the heath care provider can also request additional information through an optional custom interface. This feedback can then be viewed by the patient, thus linking communication and care from the patient and the provider on a daily basis and helping to better manage disease. The patient has a profile designated by his or her personal Microsoft HealthVault account. By signing in through this account, the patient will have the options, through the PHA to record daily activity, use the disease center to monitor specific health concerns, and use the medication center to track and record medication and nutritional supplements. On the other end, the health care provider will have his or her own profile and will be able to view the patient information and analyze information recorded by the patient.

# Operating Environment (Che Chu)

PHA operates on the specific user’s mobile device (smartphone or tablet). The device must support either Apple’s iOS mobile operating system or Android OS. Depending on Titanium SDK version support, development for both devices will be limited but will expand to most versions used on the market. We use a medical data platform developed by Microsoft called HealthVault to store the patient’s personal health condition and medical records, and then let the authorized healthcare providers access these data through their mobile device. The application indirectly communications with the HealthVault server using JSON, a text-based open standard for data exchange. Both patient-end and provider-end applications run in the same operating environment illustrated in Figure 2.

****

The operating environment includes a MS HealthVault Middle-Layer Server that acts as a communication medium between Microsoft HealthVault, PHA, and other platforms or applications such as Harvard’s SMART EMR. The Middle-Layer Server is setup and managed on Uconn campus. This server receives data input from PHA and converts the data into JSON. The server then uploads or updates the information on MS HealthVault through ASP.NET API which can then be used to generate a CCR compliant document in XML format. We will apply data security and authentication to the XML format such as encryption and digital signature.

Considering the constant need to access data on HealthVault, the Middle-Layer Server also link to a MySQL database that acts as a second repository for immediate access. Access policy control and account system will be implemented on the server. The server receives a query from PHA and then its access policy decides whether to authorize the access to a filtered or an unmodified medical data, or to reject the request according to the login information, the user’s account type (a patient or a healthcare provider), and the list of providers authorized to retrieve health records. The server will check if the data is updated each time when the PHA requests data. In a scenario which the patient updates his/her measurement on HealthVault directly using a computer, the server checks if there is inconsistency between the measurement value in PHA MySQL database and the one in HealthVault when the user switch back PHA to check the measurement. Reversely, the data will always be update by our Middle-Layer Server when a user modifies it on HealthVault through PHA. Hence, data on both sides are ensured to be updated and consistent.

**Information (M. Swircenski)**

The purpose of the PHA is to facilitate information transfer between the patient and the healthcare provider. Thus, information is an important consideration. There are many types of information used by the PHA application, divided into four groups: Daily Activity, HealthVault Information, Profile Information, and Application Settings.

## Daily Activity:

This is information entered by the patient recording their daily activities. This includes disease progress, diet and exercise, and their medication usage. The information is stored in JSON objects locally, and transmitted to the HealthVault database. The length of time each type of data is stored locally is controllable in the settings; the defaults are two weeks.

## Disease Progress:

This is information entered by the patient recording the daily progress of their disease(s). This data can analyzed locally to create charts and graphs displaying a visual representation of how the disease is progressing. The analysis is only stored temporarily, either until another analysis is run or the mobile device needs to reclaim the space. The user can, however, choose to save an analysis so that is stored permanently until the user decides to delete it.

## Diet & Exercise:

This is information entered by the patient recording their daily diet and exercise.

## Medication Usage:

This is information entered by the patient tracking their daily medication.

## HealthVault Information:

This is information used to interact with the HealthVault database. This includes information used to connect to the HV service, information regarding the identity of the application (and thus what data it can access), and session authentication.

## Connection information:

This information is used to communicate with HV. It is hard-coded into the application and not changeable by the user, and consists of URLs and/or IP addresses.

## Application Identity:

This information is used to identify the application. It is created automatically when the application is first started and connects to HV. The Patient chooses which applications are allowed to access their records. This information is kept permanently.

## Session Authentication Token:

This is information regarding an authenticated connection to the HV service. It is obtained by sending the application credentials to the HV service. Afterwards, the application only needs to use the authentication token to communicate with the service. The information lasts until HV tells PHA that the session has expired, at which point PHA repeats the above and gets a new session token.

## Profile Information:

This is information storing information about the patient and their healthcare provider(s). It consists of strings and numbers. The information is stored permanently until changed by the user.

## Patient Profile:

This information is taken from the Patient’s HV account information.

## Healthcare Provider Profile:

This information is input by the patient for the purpose of easily tracking permissions granted.

## Application Settings:

This is information regarding various local application settings and preferences. It is stored permanently, and can be changed by the user.

# Performance (Wei Lin)

The front end of the PHA system will be run using a smartphone mobile device. This kind of system offers many new technologies that can be integrated into the PHA application such as pedometers and accelerometers that can be used to collect movement data. This can be useful for collecting data for the daily activity screen such as entering data for workouts, or using the camera to take BP and pulse. The multi-touch gestures can also be integrated into the user interface to increase productivity for the user. However, the mobile system can also have challenges to consider. First, there are dozens of different smartphones out on the consumer market and every device has different hardware and software. The PHA application has to be able to detect the different hardware with different capabilities and functionalities. For example, some devices might not support multi-touch gestures so the system has to be able to adjust according to what hardware is the user using. The software is also a performance issue. The two main operating systems on the market are iOS and Android. Each of these also has different versions with different capabilities and performance. The application has to be able to perform basic tasks based on which version of which operating system it’s using. By using the Titanium platform, two sets of native source code for iOS and Android can be generated from one set of Titanium source code.

The performance of the system in the back-end is also important in order for the application to run smoothly. First, the system has to be able upload and download data efficiently through Microsoft HealthVault. An intermediate MySQL database can also be used to diversify traffic, which can be used store temporary data before being pushed to Microsoft HealthVault. Another performance factor to be considered is the limitations of using cellular network for uploading and downloading data from databases. Cellular network speeds can be difficult to predict according to the location of the user and it is usually slower than the current Wi-Fi technology. The system should also detect what kind of data speed is the current device getting before downloading and uploading new data onto the device. To limit the use of cellular connections, the application should also be able to store some data if there’s no network connection. This application should overall be able to support older technology as well as newer generation models because this application is geared towards lower income consumers that might not be able to afford the newest and best technologies.

# Security (J.R)

The aim is to store and access personal medical information for both the patient and provider without any possible security threat while following HIPAA guidelines. Communications using XML schemas, namely JSON, will therefore interact with open medical records such as MSHV and openEHR to properly enforce privacy rules which are of utmost importance.
Personal Health Assistant (PHA) has two separate applications, one intended towards the patients and the other towards the providers. The patients will be able to manage their PHR and share this information with their provider’s application while enforcing the patient security policy.

The patient oriented application will be able to store information in MSHV and give the provider special permissions to decide if they can edit or view certain information. Depending on these permissions, the provider will have limited access to the patient’s information. If the patient has multiple providers, the patient application will have the ability to give different permissions to each provider. The application will be designed with access control in mind based on the roles the users play. Certain users will have access to certain parts of the XML schemas that contain all the information and therefore ‘roles’ will be created or passed on between users. In order to achieve this, there is a security framework in place that provides customizable access to XML elements by generating XCAML security policies and enforcing security at the runtime level on XML instances to ensure data is delivered securely. PHA also uses a central server to generate reports and retrieve information from MSHV using a series of JSON API calls and XML HealthVault API.

User security policies can be as important and their personal information. They also need to be retrieved each time a call is made to the server. Hence users policies are created with their account and stored along with the PHR. The policies are filtered through first before any information is sent to the providers. Hence, the security is achieved by removing the information which should not be visible to the provider before it is sent.
MSHV service allows for more than one log in system. The user can use Windows Live ID, Facebook ID credentials, or identifiers from certain providers. This can be done directly with HealthVault’s website and the PHA app connects to this information to grant or deny access though MSHV’s API. The service will also be tied to personal information such a name, date of birth, email, address, etc.

A lot of the user permission will be handled in the back-end. However the user also needs to have a clear and easy way to set these security policies. This is done by adding a permissions page with clear options for each provider the patient is tied to. The patient can easily see all the information currently shared and check or uncheck view and edit boxes to change the permissions. Patients have all the access to MSHV at all times and can therefore change the levels of access for Custodian Access (providers), which include: Read the record; Change the record; Delete the record, and Grant and revoke to others any level of access to the record.

**Glossary (Everyone)**

**HealthVault:** Microsoft HealthVault is a platform from Microsoft to store and maintain health and fitness information

**openEHR:** An open standard specification in health informatics that describes the management and storage, retrieval and exchange of health data in electronic health records (EHRs).

**JSON:** (an acronym for JavaScript Object Notation) A lightweight text-based open standard designed for human-readable data interchange.

**Android (Operating System):** A Linux-based operating system for mobile devices developed by Google.

**iOS:** An operating system developed by Apple Inc. that is used on all of their mobile devices.
**Interfaces:** A way for the user to interact with the software/application.

**Software Development Kit (SDK):** Set of development tools that allows for the creation of applications for a specified system/environment.

**Titanium Mobile SDK:** Development tool that uses mobile operating system API’s to create native applications that perform and behave just like they were written natively.

**Database:** A structured set of data that can be organized, stored and accessed in various ways.

**HIPAA:**  The Health Insurance Portability and Accountability Act, which protects the privacy of individually identifiable health information.

**ASP.NET**: a web application framework developed by Microsoft to allow programmers to build dynamic websites, web applications, and web services. It is a successor to Microsoft’s Active Server Pages (ASP).

**API**: Application Programming Interface is a specification used as an interface by software to communicate with different components.

**CCR**: The Continuity of Care Record is a specification for a subset of patient information that encapsulates current and the most relevant facts about a patient’s health condition.

**XML**: extensible Markup language is a markup language that defines a set of rules for encoding documents in a format that is human-readable. XML files are often used as a mean to exchange data with various software platforms.

**MySQL**:  MySQL is an open source relational database management system developed by Oracle (formerly Sun) and is popularly used in web applications.